ON FRACTAL BASED REPRESENTATION OF IMAGES WITH APPLICATIONS TO IMAGE PROCESSING

Suman Kumar Mitra

Machine Intelligence Unit

Indian Statistical Institute

Calcutta 700 035, India

A thesis submitted to the *Indian Statistical Institute*
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

1999

Revised
2000

To the memory of my *Mother*

ACKNOWLEDGMENTS

I wish to express my deep gratitude to *Dr. C. A. Murthy*, my supervisor, whose unconditional encouragement, support and enthusiasm made my research work a truly enjoyable and exciting experience. He not only guided me in academic matters, but also helped me to face the challenges of life. I consider it a privilege to come in contact with a person like him having profound knowledge and yet humble.

I owe a lot to *Professor M. K. Kundu* whose constant inspiration and guidance (specially in the absence of Dr. C. A. Murthy) made it less formidable to complete this work. I am thankful for his valuable suggestions during the preparation of the manuscript and for his kind permission to include joint research works in the manuscript.

My indebtedness to *Professor S. K. Pal* is immense. His affection and the trust which he has shown to me has been a great motivation behind this work. I also extend my gratitude to him for his constructive criticism during the preparation of this manuscript.

My deep sense of gratitude is due to all faculty members, specially to Mr. S. N. Biswas, of Machine Intelligence Unit (MIU) at Indian Statistical Institute (ISI), Calcutta for their suggestions and encouragement. I also wish to thank all my friends at ISI, Mr. R. K. De, Dr. S.N. Sarbadhikari, Mr. S. Sinha, Ms. Susmita Ghosh (De), Dr. S. Bandyopadhyay, Ms. M. Acharyya, Mr. P. Mitra, Mr. S. Dutta, Ms. K. Basu and Mr. S. Saha.

A special note of thanks goes to the office staff of MIU at ISI, for their patience while dealing with my several office works and for the humorous time I spent with them.

I express my sincere thanks to the authorities of Indian Statistical Institute for the facilities extended to carry out my research work.

I take this opportunity to express my gratitude to my teachers and friends at the department of Statistics, University of Calcutta, Calcutta, India, the department of Statistics, Presidency College, Calcutta, India, St. Xaviers College, Calcutta, India and the Vivekananda Institution, Howrah, India. Teachers whose inspiration have tremendous impact in my life are Ms. K. (Chakraborty) Chatterjee, Mr. B. B. Roy and Professor T. Krishnan.

I shall forever remain indebted to my parents for helping me in all phases of life. It is beyond my capacity to express, within a few words, the amount of support I have

received from them. A deep sense of gratitude goes to my family members, specially to my brother *Sumit*, and friends for not taxing me much and for their heartfelt trust on me during the tough times. Thanks are due to my in-laws for their constant encouragement.

Finally, I would be mistaken if I do not acknowledge the support and inspiration that I received from my wife *Swati*, who always kept me away from all sorts of problems of my family.

Place: ISI, Calcutta
Date:

Suman Kumar Mitra

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Scope of the Thesis

1.1 Introduction

The language of an image is universal. Images were the means of communicating information in ancient days. Even today, although people from different parts of the world speak in different languages, an image conveys almost the same universal meaning to all. With the rapid development of modern computer technologies and with the increasing attempt in getting information at ones finger tips, the importance of communication of information using images can not be ignored.

Images are stored in computers in the form of a collection of bits representing pixels (picture elements). Pictures are to be digitized to store them in computers. The term digital image refers to a two dimensional function defined on a discrete domain and is denoted by $I(x, y)$; the value of I at spatial coordinates (x, y) is known as the pixel value and it represents the light intensity of the image at that point. Now onwards, the term image will refer to a digital image.

The representation of images with two dimensional function or a two dimensional array of pixel values is one of the basic forms widely used in image processing. Image processing usually deals with two different aspects. One is improving the image quality for human interpretation, and the other is the processing of image information for machine perception. Contrast enhancement, noise cleaning, magnification or zooming, edge extraction and segmentation are certain basic tasks of image processing.

Images can also be represented in terms of the parameters of some suitable mathematical models. Here, a class of images are modeled by a mathematical form and the parameters of the form represent the images. The most important and direct benefit of representing an image in terms of the parameters of a mathematical model lies in the reduction in the number of bits required to store it in the computer. This results in reduction of the size of the image data. The task of storing an image in some alternative form, instead of the primitive pixel form, and thereby reducing memory requirement, is called image compression. The representation of an image in the compressed form should be such that the original form of the image can be reconstructed easily whenever necessary. Some of the important application areas of image compression are image transmission and building up digital/video image libraries. The process of image compression along with decompression is an important component of image processing.

In an image compression process, the input used is a digital image and the output is its coded version. To develop an efficient image compression-decompression process, one should concentrate first upon the representation of images by a suitable mathematical model.

The present thesis deals with certain aspects of fractal based representation of images in the context of compression. This includes development of efficient algorithms and study of convergence of fractal coder. The investigation also provides results demonstrating the effectiveness of fractal representation for performing some other image processing tasks *e.g.*, image magnification and edge extraction. In this regard, efficient algorithms have been developed where the fractal codes of images are used as input. One may note that only a few attempts have been made, so far, to develop image processing operations which are directly applicable on the coded form of an image.

The usage of fractals [12, 75, 109] for representing and generating real world objects like clouds, mountains, trees, leaves and so on [12, 38, 106, 137, 139] is well documented. It is amazing how a simple fractal can model complicated real world objects. The fractal model of real life images is available in the literature [11, 78].

Before we describe the motivation and the scope of the thesis, let us explain the theory of fractals for image representation and compression, and different fractal based image processing methodologies. A brief description of the basic principles of image compression, magnification and edge extraction tasks is presented in Appendix A.

1.2 Fractals, Iterated Function System and Images

The following subsections deal with the topics which are relevant to our investigation. These include the definition of fractals, the theory of iterated function system and description of partitioned iterated function system in the context of image compression.

1.2.1 Fractals

The word fractal was first proposed by B. B. Mandelbrot [109] from the Latin word *fractus*, meaning broken. He used this term to define an object which is too irregular to be defined by classical geometry. The fractal objects, inspite of the name fractal,

have fine structures or detail in any scale. Mandelbrot [109] defined a fractal to be a set having Hausdorff dimension greater than the topological dimension. Note that the topological dimension of a set is always an integer where as the Hausdorff dimension is not necessarily an integer. The term "fractal" has been defined differently by different authors over the years [54]. We, in this thesis, won't go into the specifics of various definitions. The common factor of these definitions is the presence of some form of self similarity. It has also been found that many of the fractal objects can be defined recursively, *e.g., Cantor set, Von Koch curve, Julia set* and *Mandelbrot set.*

1.2.2 From Self Similarity to Iterated Function System

The investigation on fractals by Mandelbrot [109] opened a new horizon of science, *i.e.,* fractal geometry. The computer technologies make it possible to visualize complicated but beautiful fractal objects [137] as well as to model real world objects [12, 139]. M. F. Barnsley was one of the pioneers who modeled real life images by means of deterministic fractal objects exploiting the self similarity present in the images [11]. He suggested a fractal description of real life images in terms of the iterated function system (IFS) [12, 13]. A set of affine contractive maps on a space is called IFS on the same space. The most interesting property of IFS is that it produces a fixed point called attractor, when the maps are used recursively starting with any arbitrary set. With the help of iterated function system, along with Collage theorem [12], Barnsley laid the foundation stone of fractal based representation of an image. This fractal model of images led towards a solution to the problem of image compression [16].

In case modelling a real image by an IFS, it is enough to store the relevant parameters of the affine contractive maps instead of storing the whole image in two dimensional functional form (basic matrix form). In this way the problem of huge memory requirement for storing an image in computers can be avoided. The image compression problem using IFS can be looked upon as the inverse problem of modeling real life images by IFS. Here, the task is to find an appropriate IFS whose attractor approximates the given image.

1.2.3 Mathematical Foundation of IFS

The detailed mathematical description of IFS and related theories are given in [12, 15, 52, 55]. Some relevant definitions and theorems are stated here. Proofs of these theorems have not been stated here as they already exist in literature [12]. Here, we define an IFS starting with the definition of a complete metric space. The most important result, which is the main interest of the present thesis, is the Collage theorem. The use of invariant set of IFS for approximating images has been described in terms of the Collage theorem. Other definitions and theorems are stated for better understanding of the theorem of interest.

Definition 1.1: *Given a set X, a function $d : X \times X \rightarrow [0, \infty)$ is called a metric, if*

1. $d(x, y) = 0$ *iff* $x = y \ \forall x, y \in X$, *(identity)*

2. $d(x, y) = d(y, x) \ \forall x, y \in X$, *(isometry)*

3. $d(x, y) \leq d(x, z) + d(z, y) \ \forall x, y, z \in X$ *(triangular inequality)*

The pair (X, d) is called a metric space. ♠

Definition 1.2: *A sequence $\{x_i\}_{i=1}^{\infty}$ of points in a metric space (X, d) is called a Cauchy sequence if, for any given $\epsilon > 0$, there is an integer $N_\epsilon > 0$ such that $d(x_i, x_j) < \epsilon \quad \forall i, j > N_\epsilon$.* ♠

Definition 1.3: *A metric space (X, d) is said to be complete if any Cauchy sequence $\{x_i\}$ in X converges to some point $x \in X$.* ♠

Definition 1.4: *Let (X, d) be a complete metric space. Then $\mathcal{H}(X)$ denotes the space of all non empty compact subsets of X.* ♠

Definition 1.5: *Let (X, d) be a complete metric space. Then the Hausdorff distance "h" between A and B in $\mathcal{H}(X)$ is defined as*

$$h(A, B) = Max\{d(A, B), \ d(B, A)\}.$$

Here $d(A, B)$ is defined by

$$d(A, B) = Sup\{d(x, B) : x \in A\}.$$

Here $d(x, B)$ is called the distance from the point x to the set B and is defined by

$$d(x, B) = Inf\{d(x, y) \ : \ y \in B\}. \ \spadesuit$$

Definition 1.6: *A map $w \ : \ X \to X$ is called contractive if there exists s, $0 \leq s < 1$, such that*

$$d(w(x), w(y)) \leq s \ d(x, y) \ \forall x, y \in X.$$

Here, 's' is said to be a contractive factor of w. $\spadesuit$

Lemma 1.1: *Let w be a contractive map on the complete metric space (X, d). Then w is continuous.* $\spadesuit$

Lemma 1.2: *Let w be a contractive map on the complete metric space (X, d). Then w maps $\mathcal{H}(X)$ into itself where*

$$w(B) = \{w(x) : x \in B\}, \ \forall B \in \mathcal{H}(X).$$

w is a contractive map on $(\mathcal{H}(X), h)$ with contractivity factor s. $\spadesuit$

Lemma 1.3: *Let (X, d) be a complete metric space. Let $\{w_i \ : \ i = 1, 2, \cdots, n\}$ be contractive maps on $(\mathcal{H}(X), h)$. Let the contractivity factor for w_i be denoted by s_i for each i. Define $W \ : \ \mathcal{H}(X) \to \mathcal{H}(X)$ by*

$$W(B) \ = \ \bigcup_{i=1}^{n} w_i(B), \ \ \forall B \in \mathcal{H}(X).$$

Then W is a contractive map with contractivity factor
$s = max\{s_i \ : \ i = 1, 2, \cdots, n\}$. $\spadesuit$

Definition 1.7: *An iterated function system (IFS) consists of a complete metric space (X, d) together with a finite number of contractive maps $w_i \ : \ X \to X$, with respective contractivity factors s_i, for $i = 1, 2, \cdots, n$. The notation for this IFS is $\{X; w_1, w_2, \cdots, w_n\}$ and its contractivity factor is s where $s = max\{s_1, s_2, \cdots, s_n\}$.* $\spadesuit$

The preceding results are useful in proving the following theorem of the existence of a fixed point of IFS.

Theorem 1.1 (IFS Theorem) : *Let* $\{X; w_1, w_2, \cdots, w_n\}$ *be an IFS with contractivity factor* s. *Then the transformation* $W : \mathcal{H}(X) \to \mathcal{H}(X)$ *defined by*

$$W(B) = \bigcup_{i=1}^{n} w_i(B), \quad \forall B \in \mathcal{H}(X),$$

is a contractive map on the complete metric space $(\mathcal{H}(X), h)$ *with contractivity factor* s; *i.e.*

$$h(W(B), W(C)) \leq s.h(B, C), \quad \forall B, C \in \mathcal{H}(X).$$

Let $W^N(B)$ *is defined as*

$$W^N(B) = W(W^{N-1}(B)); \quad \forall B \in \mathcal{H}(X) \text{ and } \forall N \geq 2,$$

where

$$W^1(B) = W(B); \quad \forall B \in \mathcal{H}(X).$$

It has a unique fixed point, called **attractor**, $A \in \mathcal{H}(X)$, *which obeys*

$$A = W(A) = \bigcup_{i=1}^{n} w_i(A),$$

and is given by

$$A = \lim_{N \to \infty} W^N(B); \quad \forall B \in \mathcal{H}(X). \ \spadesuit$$

Besides the contractive maps, one may need a condensation map. The theorem of IFS is also valid when the condensation map is included in the IFS along with the contractive maps.

Definition 1.8: *Let* (X, d) *be a complete metric space and let* $C \in \mathcal{H}(X)$. *Define a transformation* $w_0 : \mathcal{H}(X) \to \mathcal{H}(X)$ *by* $w_0(B) = C$; $\forall B \in \mathcal{H}(X)$. *Then* w_0 *is called a condensation transformation and* C *is called the associated condensation set.* $\spadesuit$

The condensation map is also a contractive map with unique fixed point.

Definition 1.9: *Let* $\{X; w_1, w_2, \cdots, w_n\}$ *be an IFS with contractivity factor* $0 \leq s < 1$. *Let* $w_0 : \mathcal{H}(X) \to \mathcal{H}(X)$ *be a condensation transformation. Then* $\{X; w_0, w_1, w_2, \cdots, w_n\}$ *is called an IFS with condensation with contractivity factor* s. $\spadesuit$

Theorem 1.2 (IFS Condensation Theorem): *Let* $\{X; w_0, w_1, w_2, \cdots, w_n\}$ *be an IFS with condensation, with contractivity* s. *Then the transformation* $W : \mathcal{H}(X) \to$

$\mathcal{H}(X)$ *defined by*

$$W(B) = \bigcup_{i=0}^{n} w_i(B), \quad \forall B \in \mathcal{H}(X),$$

is a contractive map on the complete metric space $(\mathcal{H}(X), h)$ *with contractivity factor* s*; i.e.*

$$h(W(B), W(C)) \leq s.h(B, C), \quad \forall B, C \in \mathcal{H}(X).$$

Also $W^N(B)$ *is defined as*

$$W^N(B) = W(W^{N-1}(B)); \quad \forall B \in \mathcal{H}(X) \text{ and } \forall N \geq 2,$$

where

$$W^1(B) = W(B); \quad \forall B \in \mathcal{H}(X).$$

It has a unique fixed point, called **attractor**, $A \in \mathcal{H}(X)$, *which obeys*

$$A = W(A) = \bigcup_{i=0}^{n} w_i(A),$$

and is given by

$$A = \lim_{N \to \infty} W^N(B); \quad \forall B \in \mathcal{H}(X). \ \spadesuit$$

Remarks :

1. Though the definition 1.8 and the theorem 1.2 refer to a single condensation map used in the IFS, they can be generalized to have more than one condensation maps. The theorem 1.2 holds for such a case.

2. Now onwards, unless stated otherwise, IFS refers to IFS with condensation

Next we are going to state the Collage theorem for an IFS.

Theorem 1.3 (Collage theorem): *Let* (X, d) *be a complete metric space. Let* $T \in \mathcal{H}(X)$ *and let* $\epsilon \geq 0$ *be given. Choose an IFS* $\{X; w_1, w_2, \cdots, w_n\}$ *with contractivity factor* s *such that*

$$h(T, \bigcup_{i=1}^{n} w_i(T)) \leq \epsilon,$$

where h *is the Hausdorff metric. Then*

$$h(T, A) \leq \frac{\epsilon}{1-s},$$

where A is the attractor of IFS. ♠

Remarks :

1. In the theorem 1.3, the set T is approximated by the set A, and hence by the IFS $\{X; w_1, w_2, \cdots, w_n\}$. The theorem provides a method for constructing IFS and approximating the target set by the attractor of the IFS. Note that the existence of "A" does not depend on the set "B" which is a starting set for obtaining "A". Thus once the IFS of a set T on a space is obtained, an approximation of that set can be obtained starting from an arbitrary set on the same space.

2. Usually the contraction maps considered for the development of the theory in this subsection, are taken to be affine.

1.2.4 From IFS to Partitioned IFS

The theory of IFS along with Collage theorem have been described above. But, for the purpose of implementation, many authors have used Partitioned IFS (or Local IFS) for image compression. Based on partitioned IFS, a fully automated fractal image compression technique of digital monochrome images was proposed by Jacquin [78, 79, 80, 81]. He has suggested an algorithm by means of which a real life image can be approximated by a set of transformations. The technique is known as partitioned [58] or local [15] iterated function system. The partitioned/local IFS is an IFS where the domain of application of the contractive affine transformations is restricted to the small portion of the image instead of the whole image as in the case of IFS. The application of partitioned iterated function system (PIFS) technique or the methodology for finding PIFS for a given image is different from that of IFS. Generally, PIFS is considered to be a simpler extension of IFS and it is assumed that the theoretical foundation of PIFS is almost same as that of IFS. In PIFS technique, the given image is first partitioned into non-overlapping blocks and a separate transformation for each block is then found out on the basis of similarity with other blocks, located any where within the image support. The scheme is called partitioned IFS as the transformations are applied partition wise.

Intuitively, the PIFS scheme is based on self similarity present in an image, *i.e.*, under suitable affine transformations, larger blocks of the image look like smaller blocks in

the same image. The smaller image blocks are called "range blocks" and the larger image blocks are called "domain blocks" [80]. If a suitable affine transformation for a particular range block is not available using any one of the domain blocks, then a reduction in the size of that range block would be usually helpful. However if the user doesn't want to reduce the size of the range block, a condensation map of the range block would make the approximation error free. Here, a condensation map of the range block is simply the copy of the range block itself. Thus, though the notion behind PIFS scheme is self similarity, self similarity is not a prerequisite for applying PIFS scheme.

The set of affine transformations along with the selected domain block locations, provides a complete representation of the original image. Thus by storing the parameters of the affine transformations and the locations of the domain blocks, instead of the basic form (pixel matrix) of the image, a great amount of reduction in the memory requirement could be achieved. The set of affine transformations is utilized to regenerate that image, whenever necessary, starting from any arbitrary image. In the process of regeneration, some amount of loss in the image information occurs. The process of obtaining PIFS of a given image together with the regeneration of a close approximation, in terms of the attractor of the PIFS, of the original image is called the fractal image compression scheme. Figure 1.1 shows the encoding-decoding process of fractal image compression scheme. The mathematical description of the PIFS scheme will be presented in Chapter 2.

In the following section we provide a review of different methodologies based on partitioned iterated function system for digital images.

1.3 Survey of PIFS Based Methodologies in the Context of Digital Images

Since the thesis pertains to some image processing tasks like image compression, magnification and edge extraction, the basic principles of these tasks has been provided in Appendix A.

The application of PIFS, so far, is mainly restricted to perform image compression task. To our knowledge, no significant attempt, with a few exceptions, has been made

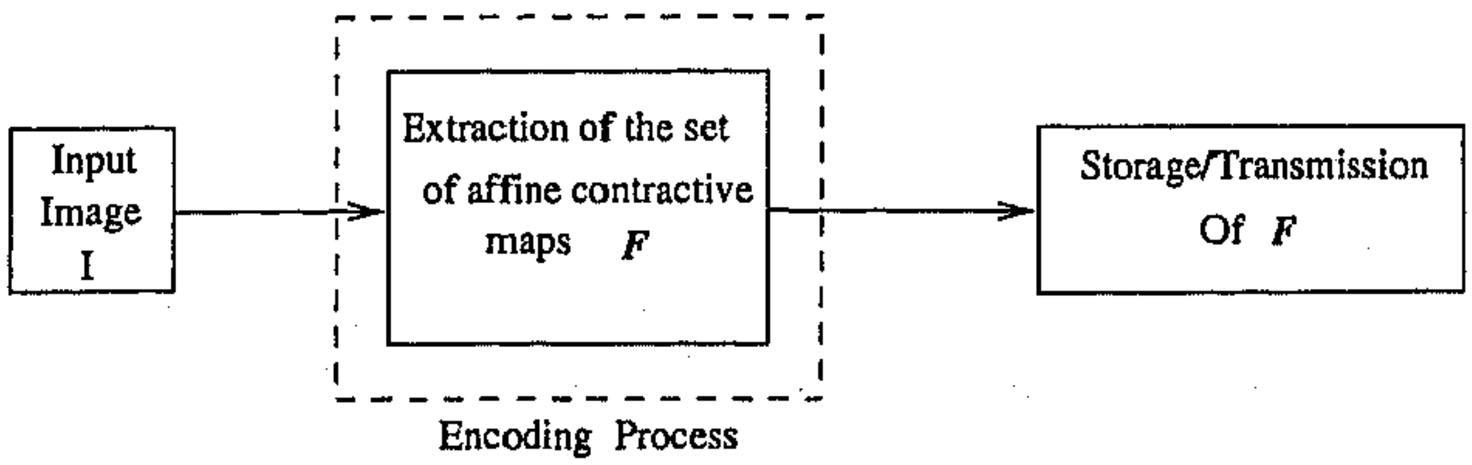


Fig. 1.1: Encoding-Decoding of fractal image compression scheme

to utilize PIFS (fractal) codes for carrying out image processing tasks other than image compression. This section includes a detailed survey of the literature of PIFS based image compression.

As mentioned earlier, the first implementable PIFS based image compression scheme was proposed by A. Jacquin [78, 79, 80]. Since then, fractal image compression has received considerable attention and most of the fractal compression schemes appeared so far in the literature are based on Jacquin's type compression technique. Most of the schemes of this kind are computationally expensive *i.e.*, the encoding time to generate PIFS codes is usually huge. The problem here is to find a suitable approximation of an image by a set of affine contractive maps. It can be looked upon as solving a set of inverse functions whose attractor approximates the given image optimally [51, 61]. Some initial solutions to this problem have been suggested by Pentland [140] and Beaumont [20]. A good collection of both theoretical and application aspects of several fractal image compression schemes is available in [56, 57].

The scheme due to Jacquin consists of finding suitable transformations defined on image blocks. These transformations are affine in nature, as approximation of smaller image blocks (range blocks) are carried out by applying transformations on suitably chosen larger image blocks (domain blocks). As the selection of appropriate domain blocks and transformations usually takes a huge computational time, he has suggested a trimming of search space following a classification scheme well known in image processing [146]. Larger image blocks are stored into a set of categories namely shade blocks, edge blocks and midrange blocks. Same classification has been employed to classify the range blocks too. For each range block, domain blocks of the same category are searched, and that domain block which suitably approximates the candidate range block under an affine contractive map, is found. The preciseness of the approximation is judged by root mean squared error (RMSE). Note that the number of parameters stored for range blocks of different classes is different [80]. Thus to indicate the class information of a range block, one needs to store another parameter.

The literature of PIFS based image compression has addressed mainly the problem of suggesting an effective algorithm for finding a set of suitable transformations, called fractal coder, which can approximate the image. In regards to this, several methodologies have been suggested with mainly three motivations namely (1) improving the performance of the fractal coder, (2) reduction in the search space for finding the suit-

able set of transformations, and (3) analyzing the convergence properties of the fractal coder.

To develop a fractal coding algorithm, proper trade-off has to be made in between the compression ratio and the quality of the decoded image. Though, in many of the techniques of fractal image compression, emphasis has been given either improvement in the compression ratio or improvement in the quality of the decoded image. In both the cases, the other issue *i.e.*, either compression or the quality of the decoded image has not been ignored. The number of transformations required to approximate the given image depends on the image complexity. For a given image, large number of transformations lead to better fidelity but poor compression ratio. To make a trade-off between compression and fidelity, Fisher et al. [58] have suggested two different approaches to encode a given image. Target of the first approach is to achieve more compression keeping the number of range blocks fixed up to a prefixed level in a quadtree partitioning or in a horizontal vertical (HV) partitioning. On the other hand, the second approach consists of partitioning the image, either in quadtree or in HV, up to that level at which the desired fidelity is achieved. In quadtree partitioning, every square partition may be subdivided into four smaller squares with one fourth the area of original square. The partitioning of images in quadtree approach can also be performed using the variance information of subimages [155]. A recursive partitioning of the image along horizontal or vertical lines is called HV partitioning. A mixture of triangular and rectangular partitioning of images has also been tried for the same purpose [44]. The other partitioning scheme used in fractal image compression is lapped partitioning [71]. It is to be noted that, besides the information regarding location of selected domain blocks and the parameters of the selected maps, the information regarding partitioning scheme is also required to be stored. This is true for all the schemes of this kind to reproduce the image whenever necessary. On the other hand, in non overlapping square partitioning, no extra bit is required to store the information regarding partitioning scheme. Thus a complicated partitioning scheme may reduce the compression ratio compared to that of square partitioning.

Fisher et al. have also pointed out another important phenomenon regarding the contractivity factor of the set of maps and coined the term eventual contractivity [58]. It has been observed that eventual contractivity is a sufficient condition to ensure the convergence of PIFS code during decoding. A generalized Collage theorem of an IFS,

in the context of eventual contractive maps, has been suggested by Fisher et al. [56].

Some block based fractal image compression techniques utilize irregular shaped blocks. In this direction, a region based partitioning scheme of the image has been suggested by Thomas and Deravi [170]. The algorithm starts with defining a seed range block among all the range blocks, and searching for appropriate domain to range transformation. The algorithm then attempts to extend the seed range block in all four principle directions. In the extended form, the parameters of the extended transformation are either same as that of seed or may be selected on an adaptive basis. The selection of fractal codes using an adaptive technique is also available in [103].

A different scheme for encoding images by PIFS using variable block sizes has been proposed by Tanimoto et al. [168]. In this scheme the range blocks are determined by a split and merge method. In this method, range blocks of various sizes and various shapes are extracted from the given image. The concept of variable size blocks has been extended by Ruhl et al., results in an adaptive partitioning of images to determine the range blocks. In this algorithm, the starting range blocks are small square blocks. Now, for the merging of the starting range blocks, an evolutionary programming [156] and its deterministic version [152] have been adopted. Another significant contribution, along this line, is the overlapped adaptive partitioning of image blocks, suggested by Reusens [150]. This overlapped partitioning is embedded in the quadtree segmentation of the image support. The region of overlap is a function of the block size at each level of the quadtree segmentation. During decoding, the value of a pixel corresponding to an overlapped region is computed as the weighted sum of the different contributions leading to that pixel.

To solve the issue of improving the quality of the decoded image, visual perception has been used in the context of searching the matched domain blocks and the transformations [33]. An entropy based constraint has also been used in this regard [17]. Unlike exploiting the similarity between range blocks and domain blocks, Rinaldo and Calvagno [151] designed an image coder in which the similarity of different sub-bands in a multiresolution decomposition of the image is exploited. In a quadtree partitioning approach, the proposed coding scheme consists of approximating blocks in one subimage, from blocks in another subimage of the lower resolution with the same orientation. The transformations defined for this purpose are similar to those used in classical fractal block coders but need not be contractive. Some other fractal based compression

techniques are available in [3, 6, 21, 31, 77, 101, 105, 126, 129, 167].

Significant contributions, in the literature of fractal image compression using PIFS, have also been made by suggesting computationally efficient algorithms. The problem of expensive computational cost has been taken care by adopting various search mechanisms for finding appropriate domain blocks and the appropriate transformations. Another solution towards the same is to restrict the selection of a domain block, as a candidate for matching, by adopting either suitable classification or clustering techniques or imposing geometric constraints. A nearest neighbour search procedure has been adopted by D. Saupe [154]. In particular, a relation between fractal image compression technique and the multidimensional nearest neighbour search has been found.

A different kind of search mechanism, known as pyramid search, has been used in fractal image compression [100, 161]. The algorithm proposed by Lin and Venetsanopoulos [100] uses a pyramid search which is embedded in a quadtree partitioning. Assuming the matching error, *i.e.*, error in approximating the range blocks, to be an independent, identically distributed (i.i.d.) Laplacian random process, the threshold sequence for the objective function to minimize the error, in each pyramidal level, is derived. The computational efficiency depends on the depth of the pyramid.

It should be noted that while suggesting a search algorithm for selecting appropriate transformations as well as appropriate domain blocks for range blocks, utmost care should be taken to reduce the complexity of the algorithm as much as possible. Moreover, a complex search procedure may be computationally expensive. In such a case, the whole encoding process may be a computation intensive procedure.

The complexity reduction of the search procedure of fractal image compression using clustering of the search space is an important issue and attracts considerable attention. Lepsoy and Oien [96] have suggested an iterative algorithm to cluster the codebook, in an adaptive way. The procedure consists of subdividing the codebook into clusters with the help of cluster centers. In the encoding process, the cluster center that most resembles the range block under consideration is found and it determines the cluster to be considered. Within this cluster, the codebook block that most resembles the range block is to be found. Clustering of domain blocks using data structures, *e.g.*, k-d trees [175], r-trees [92], continuous features [69], geometric constraints [160, 177],

are also available in the literature. Some other techniques of this kind can be found in [7, 10, 22, 29, 74].

A different technique, which utilizes archetype classification, has been suggested by Boss and Jacobs [26]. In this scheme, an archetype is generated and used in the classification of range blocks and domain blocks. An archetype of a class is that member of the class that represents best all the other members of the class. The determination of archetypes is similar to the determination of a vector quantization codebook.

Note that, to use a clustering algorithm to subdivide the codebook blocks, proper number of clusters has to be determined. Moreover, a proper function to measure the dissimilarity between blocks is to be defined. The clustering procedure also needs to be validated. Without the fulfillment of above points, the results of a clustering algorithm may not be useful for the purpose of representing a search space and hence reducing it.

Comparatively less attention has been paid towards the problem of analyzing the convergence phenomenon of the fractal coders. The algorithms for fractal image compression using PIFS have been suggested on the basis of an inherent assumption that the convergence procedure of PIFS is same as that of IFS. But, in reality, there are differences between IFS and PIFS mostly in the application domain. Thus, a study on the convergence procedure of the PIFS based fractal codes in the context of image compression needs to be carried out. Towards the study of the convergence, a generalized discrete framework has been considered by Lundheim [107] in the context of signal compression. Two different approaches for studying the convergence of affine fractal operators have been suggested by W. Skarbek. In the first approach [162], effective sufficient conditions for eventual contractivity have been presented and the notion of perceptual convergence has been introduced in the case of non contractive fractal transforms. Moreover, convergence, with fewer number of iterations, of the quantized version and the mean shifted version of the fractal transform have been studied. In the second approach [163], the convergence has been investigated by an analysis of block influence graph and pixel influence graph. The graph stability condition in pixel influence graph appears to be sufficient and necessary for convergence of selecting fractal transforms. Another graph theoretical analysis of fractal coders has been presented in [4]. Other significant contributions towards this problem have been made by Hurtgen. The convergence of fractal coder concerning a necessary and sufficient condition based on the spectral radius of the transformation matrix has been investigated by Hurtgen

and Simon [73]. The convergence of the fractal transform defined on the finite dimensional vector space for signal modelling has been reported by Hurtgen and Hain [72]. A significant contribution, considering the wavelet based framework, for analyzing block based fractal compression schemes has been made by G. Davis [40, 41, 42]. In particular, it has been shown that the fractal block coder of the form of PIFS is analogous to a wavelet subtree quantization scheme. The effectiveness of the performance of fractal block coders has been investigated in a wavelet framework. Some other works on the convergence aspects of fractal image compression are available in [46, 62, 125, 130].

There are some other methodologies which utilize fractal with other compression methodologies result in new image compression techniques. Actually this kind of encoding mechanism is better known as hybrid coding. In hybrid coding more than one approach is used in a single algorithm to result in better compression or better fidelity. A large number of contributions which utilize a fractal technique and other techniques such as vector quantization [90, 91], neural networks based coding [25, 164], wavelet based coding [45], DCT based coding [169, 178], transform coding [18] and others [149, 173, 180], are available in the literature. Comparative studies between fractal image compression method and other image compression methods has been presented in [48, 59, 98, 158]. The application area of fractal technique includes image analysis [34, 76, 83, 142], satellite imagery [179], video coding [97] and others [165]. Survey of the block based fractal image compression techniques and its application are available in [2, 81]. A detailed description along with "C" source code of one such PIFS scheme is available in [15].

There are other fractal based image compression methods which do not use the approach based on PIFS. Most significant approaches among these are given in [50, 99, 172]. The fractal image coding algorithm proposed by Vines [172] is based on an orthonormal basis approach which is a hybrid method combining principles of transform coding with those of fractal encoding. The method suggested by Dudbridge [50] begins with a non-overlapping square partition. In the encoding scheme each block is again partitioned into its quadrants and separate transformations for each quadrant are then found considering the parent partition as domain block. Moreover, the coefficients of the transformations are computed by solving a set of linear equations. An interesting investigation was carried out by Lin and Venetsanopoulos [99] by incorporating nonlinear contractive functions. The proposed method includes a decoding process which

is twice faster than the usual decoding of fractal coders. The decoded images also appear to be visually better in this decoding process. Another scheme with non iterative decoder was suggested by Kim et al. [89]. A scheme with different decoding procedure has also been suggested by Hamzaoui [67].

So far, we have discussed various methodologies of image compression based on PIFS. It has been mentioned that there are only a few methodologies available for carrying out tasks like image magnification edge extraction and segmentation. In fact, this area is much neglected. Significant results are yet to be obtained.

In the next two sections, the motivation and the scope the thesis are discussed.

1.4 Motivation of the Thesis

An effective application of fractal representation of images is found in image compression. The literature of PIFS based image compression, as mentioned earlier, has focused on three basic problems. The first problem is to determine a family of contractive maps that can be used to represent or code an image. A variety of algorithms have been explored towards the solution of this problem, as described in Section 1.3. Most of these algorithms are closely related to the technique described by Jacquin [78, 80, 81]. A serious drawback of many algorithms of this kind is the huge computational cost. The second problem is to find fast and effective algorithms for approximating a given image in terms of the attractor of a set of affine contractive maps. The third problem is to analyze the convergence properties of the set of contraction maps.

One of the main goals of the present thesis is to find an algorithm which is computationally efficient. The basic task of PIFS based fractal image compression scheme is to search for similarities in image blocks under suitable transformations. In particular the problem is to search for an appropriate "domain" block and an appropriate transformation for each "range" block such that the transformed domain block becomes a close approximation of the range block. To find the best matched domain block as well as best matched transformation, one has to search for all possible domain blocks and transformations. This search mechanism makes the algorithm expensive in the sense of computational time. There are several algorithms where this issue has been considered. Most of the algorithms deal with reduction in the search space either by

adopting suitable partitioning schemes or by clustering the search space. The basic problem of adopting a partitioning scheme is that it is required to store the relevant information regarding the partition, causing reduction in the compression ratio. On the other hand, in most of the clustering techniques a proper function needs to be defined to measure the similarity of blocks forming a cluster. This may lead to an increase in computational complexity. Therefore, to make the encoding of PIFS image compression scheme computationally efficient, it is necessary to provide an algorithm where the search mechanism is fast and effective.

The present thesis addresses this problem by suggesting an algorithm where the search mechanism has been performed by Genetic Algorithms (GAs). GAs are biologically motivated random optimization techniques. One of the most important features of the GA based optimization techniques is their stochastic nature which makes these techniques efficient compared to simple random search mechanism. When the search space is discontinuous, multi dimensional and multimodal, then GAs have been found to outperform both gradient descent method and various forms of random search [8]. The solutions which are far away from the optimal are thrown away in bulk and this makes GAs to perform optimization rapidly. A detailed description of Genetic Algorithms has been presented in the next chapter. The GA based technique finds the appropriate domain block and the transformation, for a range block, from the entire search space. At the same time it is computationally efficient when compared to the exhaustive search.

One may note that an image can be considered as a two dimensional signal. So, the effectiveness of fractal for representing signals which are not two dimensional needs to be investigated. With this aim in mind, we have carried out investigation for representing one dimensional signal (curve). In this context, an algorithm for EEG signal compression has been developed. It is to be noted that there exist only a few works towards modeling [182] and coding [108, 131] of one dimensional signals.

Once the effectiveness of the fractal representation is established for image compression, the applicability of this representation for performing other image processing tasks needs to be investigated. This investigation has been carried out and it results in suggesting two different image processing operations, namely, magnification and edge extraction. There exist hardly any attempt, with a few exceptions [34, 76, 141], to explore this particular approach in the context of PIFS based technique.

Image processing operations, in general, are carried out on images, but not on the codes of the images. Hence, the code of an image needs to be decoded initially before taking up any image processing task. It may be, therefore, better computationally, if the processing could be done on the code of an image directly instead of the reconstructed image thus saving time. This approach can be regarded as compressed domain image processing. Compressed domain image processing is still in a nascent stage. Two algorithms are being suggested here in this regard. One is image magnification and other is image edge extraction.

1.5 Scope of the Thesis

The thesis deals with fractal based representation of images with its applications to image processing. The problem of image compression using a fractal model is the prime area of interest of the thesis. The fractal code which is generated in the process of image compression has been termed as the fractal representation of the image. A new method for obtaining fractal code using Genetic Algorithms and thereby reducing the computational time has been discussed. The convergence process of fractal codes has also been analyzed. On the basis of the convergence procedure of the fractal code, a few image processing tasks such as image magnification and image edge extraction have been performed. All these proposed mechanisms may lead to a new image processing system where the input is the fractal representation of the image alone. The basic advantage of such an image processing system is the reduction in computational time and memory requirements. The use of fractal model to encode colour images and curves has also been discussed. Last but not the least a new probabilistic approach for fractal image compression has been proposed. This proposed methodology is extremely fast for encoding images. The results of the investigations are summerised below on the basis of chapter headings.

1.5.1 Image Compression Using Genetic Algorithms [121, 120, 118, 123, 124]

A serious drawback of the most of the PIFS based image compression algorithms is the huge computational time. The problem of finding fast and effective algorithms to obtain the fractal code of a given image using PIFS technique has been addressed in Chapter 2. The new method for fractal image compression uses Genetic Algorithms with elitist model.

The encoding process of the PIFS technique consists of approximating the small image blocks, called range blocks, from the larger blocks, called domain blocks, of the same image, through some transformations. In the process, separate transformations for each range block are obtained. Thus the basic task is to find an appropriate domain block as well as an appropriate transformation for each range block. The information concerning the suitable domain block and transformation is called the fractal code of the range block under consideration. This information for each range block and thereby for the entire image is enough to reproduce an approximation of the given image. The whole problem of finding appropriate domain blocks and transformations can be viewed as a search problem. The present method utilizes Genetic Algorithm that greatly reduces search time for computing fractal code. Genetic Algorithms (GAs) are highly adaptive search and machine learning processes based on a natural selection mechanism of biological genetic system. GAs help to find the global near optimal solution without getting stuck at local optima as it deals with multiple points simultaneously. GAs attempt to find near optimal solutions without going through an exhaustive search mechanism. GAs perform well for optimization problems since near optimal solutions are usually satisfactory and, intuitively, the solutions which are far away from the optimal are thrown away in a bulk.

The methodology has been implemented on several real life images and the performance is found to be satisfactory in the sense of compression ratio and the quality of the decoded image. A new classification scheme for range blocks which is found to be effective for achieving better compression ratio has also been proposed. An analytical study of the proposed method has also been presented. Comparison with other fractal based image compression methods has also been reported here. It has been found that reduction in the search space can be achieved at the order of 20, if Genetic Algorithms

are used, instead of exhaustive search, for finding fractal code of a given image.

The method has also been implemented in another commonly available format of images which is 24 bits/pixel (bpp). This type of format usually represents a colour image. In the 24 bpp format, an image is made up of red, green and blue subimages, each subimage being represented with 8 bits/pixel. A standard approach for encoding 24 bpp colour images is to transform the images from the RGB representation to the YIQ (luminance and chrominance) representation. This approach has been exploited to find the fractal codes of 24 bpp colour images. A relatively large compression has been achieved in this case.

The GA based fractal based image compression algorithm is then suitably modified for one dimensional signal compression. The use of fractal technique for modeling [182] and coding [108, 131] of one dimensional signal has not been explored much. One may note that digital images can also be regarded as two dimensional signals. The objective is to investigate whether the said methodology can lead to efficient encoding of curves which are seemingly irregular in nature. As in the case of image compression, the main task here is to find the appropriate set of transformations whose attractor approximates the given signal. Out of the several possible transformations only a few affine transformations have been defined using the similarity present in the given signal. Implementation of GAs as a search mechanism provides fast encoding of one dimensional signals through PIFS.

The proposed methodology has been tested with several EEG (Electroencephalography) signals. EEG reflects the electrical activity of the brain during various states of sleep and wakefulness. The EEG signals are quite complex in nature. EEG compression can help (*a*) to augment the storage capacity of collected EEG data for later evaluation or comparison, (*b*) to facilitate transmitting real-time EEG signals to distant places and (*c*) to transmit rapidly and economically off-line EEG data over telecommunication networks to remote interpretation centers.

In particular, we have tested our method on three states of EEG (wakeful, REM and slow wave sleep). In each state a data (of four seconds' duration) consists of 1024 points. We have examined the performance of the algorithm for two level (parent and child) partition scheme as well as single level partition scheme using eight isometric transformations which are designed to encode one dimensional signals. The results are

compared with those of the existing techniques [70, 144, 171] of encoding EEG signals.

The values of cross-correlation, PSNR and SNR of the original and decoded signals have been computed to judge the performance of the proposed coding methodology. The results obtained are also found to be satisfactory clinically. The compression ratios indicate that at least 85% reduction is achieved in all the data sets. This establishes that PIFS with GAs can lead to efficient and reliable compression of EEG signals.

1.5.2 Image Magnification [117, 119]

Image magnification is a process which increases image size, keeping all the image details unaffected, in order to highlight implicit information present in the image, not evident as such. Generally, the image is represented in the form of a two dimensional array of pixels values (matrix form) and it requires huge memory space. The storage requirement is greatly reduced by using some form of image compression. But, for performing image processing tasks, the coded image is usually brought back to its normal form. It is increasingly the case that the coded form of the image is used, instead of the normal form, as the input to perform different image processing tasks. With this aim in mind, an attempt is made in Chapter 3 to develop a new magnification technique which can be applied directly on the coded form of the image. In particular, the method uses the coded form of the image as obtained by the compression technique of Chapter 2.

Conventional magnification techniques are basically interpolation methodologies which are based on linear, bilinear, cubic or bicubic interpolations. In our algorithm magnified version of an image is obtained using the reconstruction of the fractal code of that image. No magnification operator like interpolation is needed here. Only the fractal code, or, in other words, the set of affine contractive maps is needed to magnify the image. A technique whose magnification orders are powers of two has been presented. This can be extended to the case of magnification by any order.

Two distortion measures to judge the performance of the magnification algorithms have also been formulated. The purpose is to measure the distortion or similarity between the given image and the magnified image. The said techniques are image size independent and utilize both global and local information. The first one is a distortion measure,

based on the edge images and indicates the influence of artifacts like blocking, blurring and ringing which may appear due to magnification. The other is a similarity measure based on the just noticeable difference (JND) which is nothing but change in luminance of an object pixel with respect to its background pixels. Comparison with one of the most popular magnification techniques, the nearest neighbor technique, is also made.

1.5.3 Image Edge Extraction in the Compressed Domain [115, 122, 116, 120, 114]

Some form of data compression is needed to store large volume of images. Presently, it is increasingly the case that instead of using raw digital images, the compressed form is being used for performing image processing tasks. The compressed forms of images are directly accessed from the storage space. Keeping this in mind, an edge extraction algorithm in the compressed domain using fractal reconstruction has been developed in Chapter 4. Conventional edge extraction algorithms are called edge detection as edges are detected from digital images using a convolution operation defined on kernels of different sizes and having different weights based on the application demand. On the contrary there is no such kernel used in our method. Here, edges are extracted indirectly during the reconstruction of images from their respective PIFS codes.

In PIFS based image coding, pixel values are approximated by linear continuous maps in an iterative process. Since, edge pixels are formed due to discontinuities present in gray values, the process of approximating discontinuities by continuous maps would be slow. During decoding, *i.e.*, during convergence of PIFS codes, it has been found that edge pixels take more iterations to converge compared to non edge pixels. To extract edges from PIFS code, a penalty function, indicating the status of convergence, on the pixels during the process of convergence has been introduced. The penalty values of edge pixels would be more than that of non edge pixels. After a sufficiently large number of iterations, instead of storing fixed stable points for each pixel, the respective penalty values have been stored. Thus after reconstruction instead of getting an image which is very close to the original, an edge image is obtained.

The edge image thus obtained may possess some discontinuities. To overcome this, a simple edge linking scheme has also been suggested. The performance of the edge extraction system has been demonstrated on a synthetically generated image as well as

on real life images.

1.6 Conclusions and Scope for Further Research

The concluding remarks with scope for further work are presented in Chapter 5.

Chapter 2

Image Compression Using Genetic Algorithms

2.1 Introduction

This chapter discusses a new method for fractal image compression using Genetic Algorithms with elitist model. The technique described here utilizes Genetic Algorithms that greatly decrease the search space for selecting suitable fractal codes for small image blocks. This chapter presents implementation and analytical study of the proposed method along with a simple classification scheme for gray scale images. Comparisons with other fractal based image compression methods have been reported. The performance of the proposed method has also been investigated for 24 bits/pixel colour images and for one dimensional signals (curves) such as EEG signal.

The theory of image coding using iterated function system (IFS) was first proposed by Barnsley [12]. He modeled real life images by means of deterministic fractal objects *i.e.* by the attractors evolved through iterations of a set of contractive affine transformations. With the help of iterated function system, along with Collage theorem, Barnsley laid the foundation of the fractal based image compression. A set of contractive affine transformations [iterated function system (IFS)] can approximate a real image and so, instead of storing the whole image, it is enough to store the relevant parameters of the transformations reducing memory requirements. The basic problem of fractal based image compression is to find appropriate parameter values of transformations whose attractor is an approximation of the given image.

The first practically implemented fractal based image compression technique for digital image was proposed by Jacquin [80]. The encoding process consists of approximating the small image blocks, called range blocks, from the larger blocks, called domain blocks, of the image, through some operations. In the encoding process, separate transformations for each range block are obtained. The scheme also uses the theory of vector quantization [146] to classify the blocks. The set consisting of these transformations, when iterated upon any initial image, will produce a fixed point (attractor) which approximates the target image. This scheme is known as partitioned iterated function system (PIFS). A detailed description of existing methodologies of PIFS based image compression techniques has already been provided in Chapter 1, Section 1.3. The fractal techniques of image compression suffer from the non symmetry of the encoding/decoding process. The decoding step is very fast while a long execution time is necessary in the encoding step. A Genetic Algorithms based method, proposed here,

may appears to be a suitable solution to this problem. Genetic Algorithm (GA) has been used here to speed up the search for appropriate domain blocks for range blocks. We have investigated the performance of the GA based technique using two very specific range block partitioning schemes *viz.*, single level non-overlapping and two level non-overlapping. Though, GA can also be used for the same purpose for other partitioning schemes as mentioned in Chapter 1.

Genetic algorithms (GAs) [27, 43, 64, 112, 60, 134, 135] are mathematically motivated search techniques which try to emulate biological evolutionary processes to solve optimization problems. Instead of searching one point at a time, GAs use multiple search points. GAs attempt to find near optimal solutions without going through an exhaustive search mechanism. Thus GAs can claim significant advantage of large reduction in search space and time. Genetic algorithms with elitist model are used in the present case for finding the appropriate domain block as well as the appropriate transformation for each range block.

Basic features of IFS are outlined in Section 2.2. The principle and features of GAs are described in Section 2.3. The method of selecting fractal codes for a given image using GAs is described in Section 2.4. Section 2.5 presents implementation and results of the proposed algorithm. In Section 2.6, analytical study of the proposed method and comparison with the other fractal based methods are described followed by discussion and conclusions in Section 2.9. The application of the GA based schemes for coding of colour images and curves are presented in Section 2.7 and 2.8 respectively.

2.2 Image Coding By IFS

The detailed mathematical descriptions of the IFS theory, Collage theorem and other relevant results are available in Chapter 1, Section 1.2.3. Only the salient features of image coding through IFS are given below.

Let I be a given image which belongs to the set X. Generally X is taken as the collection of compact sets. For the sake of brevity and ease of notation the $\mathcal{H}(X)$ of Chapter 1 is referred as X here. Our intention is to find a set $\mathcal{W}$ of affine contractive maps for which the given image I is an approximate fixed point. Let $w_i, i = 1, 2, \cdots, n$

be n contractive maps with contractivity factors $s_1, s_2, \cdots, s_n$ and $\mathcal{W} = \bigcup_{i=1}^{n} w_i$ is such that the distance between the given image and the fixed point (attractor) of $\mathcal{W}$ is very small. The attractor "A" of $\mathcal{W}$ is defined as follows :

$$\lim_{N\to\infty} \mathcal{W}^N(J) = A, \quad \forall J \in X,$$

and $\mathcal{W}(A) = A$, where $\mathcal{W}^N(J)$ is defined as

$$\mathcal{W}^N(J) = \mathcal{W}(\mathcal{W}^{N-1}(J)), \text{with}$$

$$\mathcal{W}^1(J) = \mathcal{W}(J), \quad \forall J \in X.$$

Also $\mathcal{W}$ is contractive which is defined as follows:

$$d(\mathcal{W}(J_1), \mathcal{W}(J_2)) \leq s\ d(J_1, J_2);\ \forall J_1, J_2 \in X \quad \text{and} \quad 0 \leq s < 1, \qquad (2.1)$$

$$\text{where, } s = max\{s_1, s_2, \cdots, s_n\}.$$

Here d is called the distance measure and s is called the contractivity factor of $\mathcal{W}$.

$$\text{Let } d(I, \mathcal{W}(I)) \leq \epsilon \qquad (2.2)$$

where ϵ is a small positive quantity. Now, by Collage theorem [12], it can be shown that

$$d(I, A) \leq \frac{\epsilon}{1-s} \qquad (2.3)$$

where A is the attractor of $\mathcal{F}$.

From equation (2.3) it is clear that, after a sufficiently large number(N) of iterations, the set of affine contractive maps $\mathcal{W}$ produces a set which belongs to X and is very close to the given original image I. Here $(X, \mathcal{W})$ is called iterated function system and $\mathcal{W}$ is called the set of fractal codes for the given image I.

2.3 Basic Principles and Features of Genetic Algorithms

Among different optimization techniques that are found in the literature, the exhaustive search technique and the random search technique are some of the widely used

techniques. Exhaustive search conceptually is very simple and involves evaluation of every point of the search space. This method obviously becomes unrealistic for very large search spaces. The search space is investigated at random in a random search mechanism.

Genetic Algorithms (GAs) [27, 43, 64, 112, 60, 134, 135] are efficient, adaptive and robust search and optimization processes. GAs use the techniques of random search guided by some probabilistic rules. These processes are capable of performing well for a very large, complex and multimodal search space. The search processes in GAs are based on the notion of selection mechanism of natural genetic system [64]. GAs help to find the global near optimal solution without getting stuck at local optima as they deal with multiple points (spread all over the search space) simultaneously. To solve the optimization problem, a GA starts with the structural representation of a parameter set. The parameter set is coded as a string of finite length called the chromosome. Usually, the chromosomes are strings of 0's and 1's. If the length of string (chromosome) is L then the total number of possible strings is 2^L. A GA typically consists of the following components: [64]

- A population of binary strings.
- Objective function, usually called fitness function, and the corresponding evaluation criterion.
- Selection/reproduction procedure.
- Genetic operators and corresponding probabilities to perform genetic operations.

A detailed description of GAs and relevant theories are available in [8]. Components of GAs and the techniques for using these components are now briefly described.

Fitness function :- Usually, a function "fit" is defined on the set of chromosomes (strings) called "fitness function". It denotes the fitness value of a string. The problem here is to find the string (chromosome) which provides optimal fitness value among all strings (chromosomes). A string with high fitness value is called a good string if the optimization process under consideration is a maximization problem. On the contrary, for a minimization problem, a string with low fitness value is called a good string.

Thus, on the basis of the fitness values of the string, the best string can be searched out. Note that in this chapter, we are dealing with a minimization problem. ♠

Initial population :- Out of all possible 2^L strings, initially a few strings [say S number of strings] are selected randomly and this set of strings is called initial population [64]. In our investigation, we have taken S to be an even number though S can be an odd number too. ♠

Genetic operators :- Three basic genetic operators, i)Selection, ii)Crossover and iii)Mutation are exploited in GAs. The elitist model of GAs which keeps track of the best string obtained so far has also been used. The genetic operators are applied on the initial population to generate a new population of same size (S). Same operators are again applied on this new population to give rise to another population. The process of creation of a new population from the existing one is named as iteration. The description of the operators is given below.

Selection :- Selection is a simulation of natural selection mechanism. There are several ways of performing this selection operation. A procedure, which has been followed in this article, has been described below.

In this operation, a mating pool of strings of the current population is selected using the fitness values of strings in the population. The probability of selection of a string in the population to the mating pool is inversely proportional to its fitness value. A string with lower fitness value has a higher number of replications in the mating pool. This type of selection is called proportional selection strategy [64]. The size or the number of strings in a mating pool is taken to be same as that of the size of the initial population. ♠

Crossover :- There are several ways of performing the crossover operation [43] among the strings in the mating pool. The single point crossover operation, which has been followed in this article, has been described below. The crossover probability is represented by P_{cross}.

In this operation, $\frac{S}{2}$ pairs of strings are formed randomly from the population in the mating pool, of size S, so that every string in the mating pool belongs to exactly one pair of strings. A random number *rand* in the range [0, 1] is then generated for each pair of strings. If $rand \leq P_{cross}$; then crossover is performed. Otherwise the operation is not performed. Each pair of strings undergoes crossing over in the following manner.

An integer position k is selected randomly between 1 and $L-1$ ($L \geq 2$, is the string length). Two new strings are then created by swapping all the characters from position $k+1$ to L of the old strings. For example, let,

$$a = \alpha_1 \ \alpha_2 \ \alpha_3 \ \cdots \ \alpha_k \ \alpha_{k+1} \ \cdots \ \alpha_l$$

$$b = \beta_1 \ \beta_2 \ \beta_3 \ \cdots \ \beta_k \ \beta_{k+1} \ \cdots \ \beta_l$$

be two strings (called, parents) forming a pair for crossover operation. The strings (called, children or offspring) generated after the crossover operation are

$$a' = \alpha_1 \ \alpha_2 \ \alpha_3 \ \cdots \ \alpha_k \ \beta_{k+1} \ \cdots \ \beta_l$$

$$b' = \beta_1 \ \beta_2 \ \beta_3 \ \cdots \ \beta_k \ \alpha_{k+1} \ \cdots \ \alpha_l.$$

Usually a high value is assigned for the crossover probability P_{cross}. ♠

Thus, after crossover operation, the number of strings obtained is same as the population size S. ♠

Mutation :- In this operation every bit of every string is flipped (i.e. 0 by 1 and 1 by 0) with probability P_{mut}. P_{mut} is called the mutation probability.

One of the commonly used conventions is to assign a very small value to the mutation probability P_{mut} and keep the P_{mut} fixed for all the iterations i.e. the value for P_{mut} is independent of the number of iterations. A different strategy has been adopted, for the present case, in assigning the value for P_{mut}. We have prefixed the number of iterations of GA apriori and varied the mutation probability with the number of iterations. The mutation probability is taken to be a high value in the first few iterations. It is then gradually decreased to a smaller value up to a certain iteration. It is then again gradually increased to a higher value in a predefined fashion (Section 2.5). The process of increase and decrease of mutation probability is repeated for a few number of times. This varying mutation probability scheme has already been applied successfully in connection with an application of GAs to a pattern recognition problem [9].♠

Elitist model :- In the elitist model [112] of GAs, the knowledge about the best string obtained so far is usually preserved within the population. For this purpose, the worst string in the present population is replaced by the best string of the previous population.♠

Description of the algorithm :- The genetic algorithm is implemented using the following steps.

1. Generate an initial population Q of size S and calculate fitness value of each string S of Q.

2. Find the best string S_{bst} of Q. If the best string is not unique, then call any one of the best strings of Q as S_{bst}.

3. Construct a mating pool using proportional selection strategy (S_{bst} belongs to Q). Perform crossover and mutation operations on the strings in the mating pool and obtain a population Q_{tmp}.

4. Calculate the fitness value of each string of Q_{tmp} and replace the worst string of Q_{tmp} by S_{bst}. Rename Q_{tmp} as Q.

5. Go to step 2.

Note that steps 2, 3 and 4 together make an iteration.♠

Stopping criterion :- It has been shown that the GA would result in the optimal solution as the number of iterations in the above model goes to infinity [23]. But, in practice, the process in GA needs to be stopped after finitely many iterations. Usually two stopping criteria are used in genetic algorithms. In the first, the process is executed for a fixed number of iterations and the best string, obtained so far, is taken to be the near optimal one. While in the second criterion, the algorithm is terminated if no further improvement in the fitness value for the best string is observed for a fixed number of iterations, and the best string obtained so far is taken to be the near optimal one. In the present article, the number of iterations, say T, is fixed apriori for the termination of GAs. ♠

Genetic parameters :- The values for the genetic parameters are to be provided in order to carry out a search process. The genetic parameters are as follows:

1. String length (L).

2. Population size (S).

3. Number of iterations (T).

4. Crossover probability (P_{cross}).

5. Mutation probabilities (P_{mut}).

Exact values of these parameters for the image compression problem are given in Section 2.5 ♠

In the next section, the methodology of construction of fractal code using GAs has been described.

2.4 Proposed Methodology

2.4.1 Construction of Fractal Codes

Let, I be a given image having size $z \times z$ and the range of gray level values be $\{0, 1, 2, \cdots, l-1\}$. Thus the given image I is a subset of $\mathbb{R}^3$. The image is partitioned into n non overlapping squares of size, say $b \times b$, and let this partition be represented by $\mathcal{N} = \{\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_n\}$. Each $\mathcal{R}_i$ is named as range block. Note that $n = \frac{z}{b} \times \frac{z}{b}$. Let $\mathcal{M}$ be the collection of all possible blocks of size $2b \times 2b$ and let $\mathcal{M} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m\}$. Each $\mathcal{D}_j$ is named as domain block with $m = (z - 2b + 1) \times (z - 2b + 1)$.

Let

$$\mathcal{F}_j = \{f : \mathcal{D}_j \to \mathbb{R}^3 \; ; \quad f \text{ is an affine contractive map}\}.$$

Now, for a given range block $\mathcal{R}_i$, let $f_{i|j} \in \mathcal{F}_j$ be such that

$$\rho(\mathcal{R}_i,\ f_{i|j}(\mathcal{D}_j)) \ \leq\ \rho(\mathcal{R}_i,\ f(\mathcal{D}_j)) \ \ \forall f \in \mathcal{F}_j,\ \forall j.$$

Here ρ is a suitably chosen distance measure.

Now, let k be such that

$$\rho(\mathcal{R}_i,\ f_{i|k}(\mathcal{D}_k)) \ =\ \min_j\{\ \rho(\mathcal{R}_i,\ f_{i|j}(\mathcal{D}_j))\ \} \tag{2.4}$$

Also, let $f_{i|k}(\mathcal{D}_k) = \widehat{\mathcal{R}}_{i|k}$.

Our aim is to find $\{f_{i|k}, \mathcal{D}_k\}$ for each $i \in \{1, 2, \cdots, n\}$ such that $f_{i|k}(\mathcal{D}_k)$ approximates $\mathcal{R}_i$. Here the set $\{(f_{i|k}, \mathcal{D}_k);\ i = 1, 2, \cdots, n\}$ is called the PIFS or fractal code of image I. This code is used for decoding the given image I from any arbitrary starting image I_0.

To find the best matched domain block as well as the best matched map, one has to search all possible domain blocks with the help of equation (2.4). The affine contractive map $f_{i|k}$ is constructed in two steps. The first step is transformation of rows and columns from domain blocks to range blocks. This part is nothing but the change of co-ordinates in the two dimensional geometry. This can be achieved by using any one of the eight transformations (refer to Figure 2.1) on the domain blocks. These eight transformations are all isometric transformations since they preserve distances [80]. Once the first part is obtained, second part is an estimation of pixel values of a range block from the pixel values of the transformed domain block. These estimates can be obtained by fitting a straight line using the method of least squares. The method of least square analysis is as follows:

Let $(d_i,\ r_i), i = 1, 2, \cdots, n$ be n given ordered pairs. We like to estimate r_i's from d_i's using an equation

$$r_i = a\ d_i + b \quad \forall i = 1, 2, \cdots, n$$

. Note that there may not exist a and b such that $r_i = a\ d_i + b\ \forall i = 1, 2, \cdots, n$. Then one can find the 'best' values for a and b by minimizing

$$\sum_{i=1}^{n} (r_i - a\ d_i + b)^2.$$

The values, thus found for a and b are denoted by $\hat{a}$ and $\hat{b}$ respectively where

$$\hat{a} = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(r_i\ d_i\ -\ \frac{1}{n}\sum_{i=1}^{n} r_i\ \times\ \frac{1}{n}\sum_{i=1}^{n} d_i \right)^2}{\frac{1}{n}\sum_{i=1}^{n}\left(d_i^2\ -\ \frac{1}{n}\sum_{i=1}^{n} d_i \right)^2}$$

$$\hat{b} = \frac{1}{n}\sum_{i=1}^{n} r_i\ -\ \hat{a}\ \frac{1}{n}\sum_{i=1}^{n} d_i.$$

It is to be noted that the method of least squares has been adopted by the other researcher [49, 58] in the context of fractal image compression.

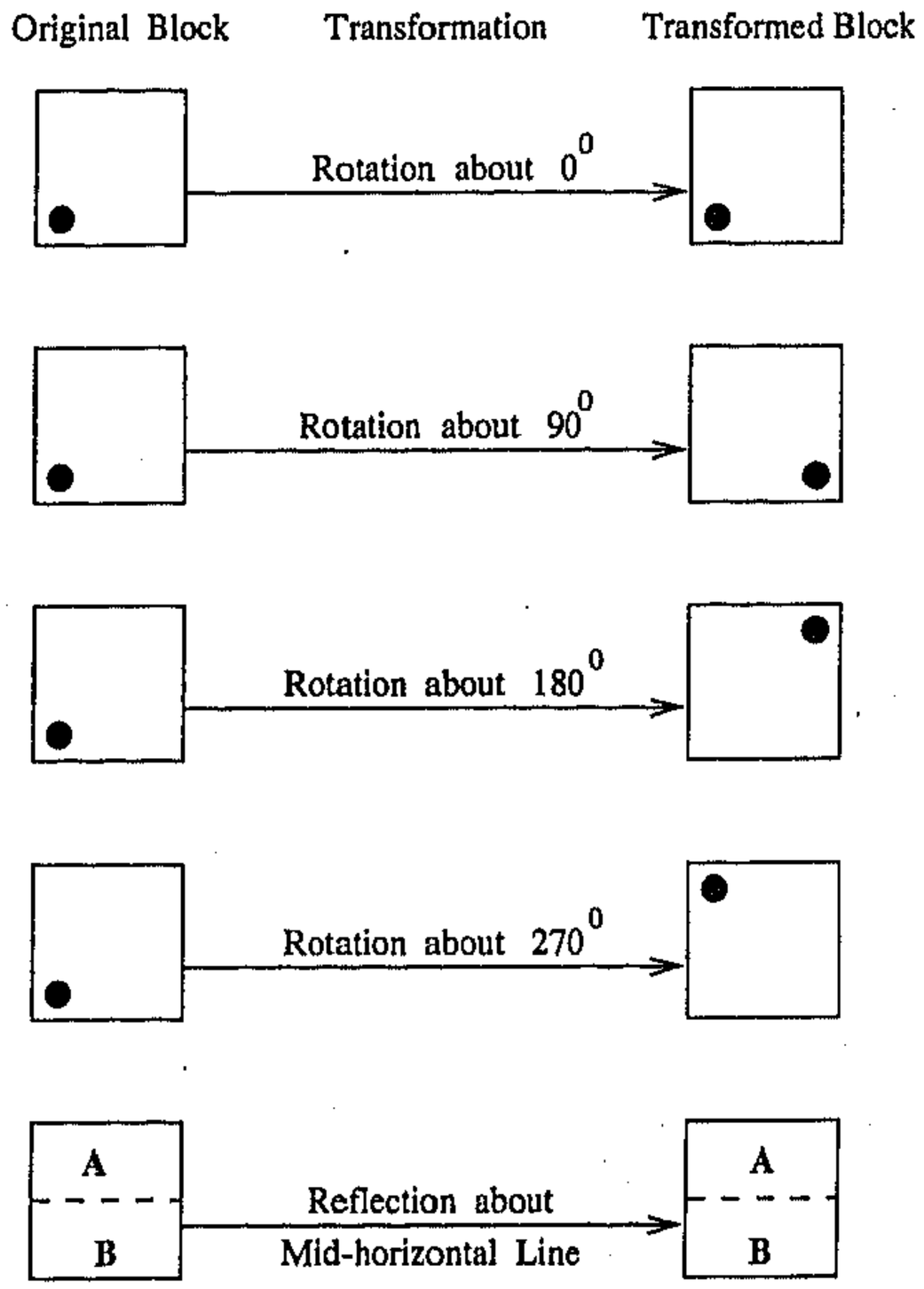


Fig. 2.1: Schematic diagram of eight isometric transformations

The distance measure "ρ" [used in equation (2.4)] is taken to be the simple root mean square error (RMSE) between the original set of gray values and the obtained set of gray values of the concerned range block. As selection of fractal code for a range block is dependent only on the estimation of pixel values of that block, it is enough to calculate only the distortion of the original and estimated pixel values of the block. Thus, RMSE is taken as the distance measure. Note that the same measure or some other forms of it had been used in other articles too [80, 170].

The performance of the PIFS scheme very much depends on the selection of appropriate domain block and transformation for a range block. The selection of appropriate domain block and transformation again depends on the value of the distance measure "ρ". A transformation and a domain block is taken to be suitable if the value of the distance measure "ρ" is very low. Now, if an appropriate transformation from a domain block to a range block is not available (*i.e.*, the distance between the pixel values of the range block $\mathcal{R}_i$ and $\widehat{\mathcal{R}_{i|k}}$ is very high), a reduction in the sizes of the range block and domain block would make the then best transformation more suitable.

Reduction in the size of the range block would increase the cardinality of the corresponding search space. Hence, the selection of an appropriate transformation would be more likely. Moreover, the distributional complexity of the pixel values of a smaller range block would be less compared to that of a larger range block. In such a case, the approximation of pixel values by a continuous affine map would be more amenable. For example, for an image of size 256×256, if the range block size (b) is 8×8, then cardinality of the search space would be $241 \times 241 \times 8$ and there will be 64 pixel values which are to be approximated by a linear continuous map. On the other hand, for a range block of size 4×4, cardinality of the search space would be $249 \times 249 \times 8$ and in this case only 16 pixel values are to be approximated. Thus, in the process of reduction in the size of the range blocks and domain blocks, range blocks are estimated more appropriately by linear continuous maps. In particular, if the range block size is 1×1, then for every domain block of size 2×2, there would be a linear transformation such that the exact pixel value of the concerned range block can be reproduced. However, if range block size is too small, then practically no compression would be achieved.

To make a trade off between amount of compression and quality of the decoded image, a two level partitioning scheme for the range blocks has been utilized in the present algorithm. The scheme is similar to the one proposed by Jacquin [80]. This two level

partition scheme takes care of approximating smaller range blocks in case of improper approximation of larger range blocks.

Another way of getting exact estimates of range blocks, in the case of improper approximation by affine maps, is the introduction of condensation maps of the range blocks. A condensation map can be of two types 1) a map which reproduces the range block as it is, and 2) a map that replaces each pixel value of a range block by the average of its pixel values. These two types of condensation maps can also be looked upon as the contractive maps with contractivity factors zero.

The first type of condensation map of the range block is simply the duplication of the range block itself. In the process of decoding, a condensation map reproduces the concerned range block as it is. Hence, the condensation map makes the approximation error free albeit without compression. However a condensation map need not always result in zero compression. If the variability of pixel values of a range block is insignificant then it could be considered that pixel values of that particular range block are almost equal. Now, if each pixel value of the range block is replaced by the average of pixel values, the error in approximating the pixel values would be insignificant. This process could be considered same as that of a condensation mapping and is called second type of condensation. This operation has been incorporated in the present algorithm with the help of a simple classification scheme.

In the proposed algorithm, GA is used to search for an appropriately matched domain block as well as an appropriate transformation for range blocks in a two level partitioning scheme. Range blocks are first categorized into one of the two classes and GA is applied to find the code of a particular type of range block. The classification of range blocks is described below.

2.4.2 Classification

The purpose of block classification is two fold. One purpose is to store fewer number of bits or to get higher compression ratio and the other is to reduce the encoding time. A simple classification scheme on range blocks is used here. Range blocks are grouped into two sets according to the variability of the pixel values in these blocks. If the variability of a block is low, *i.e.*, if the variance of the pixel values in the block is below a fixed

value, called threshold, we call the block as smooth type range block. Otherwise we call it a rough type range block. The threshold value which separates the range blocks into two types is obtained from the valley in the histogram of the variances of pixel values within blocks. After classification, GA based encoding is adopted for rough type range blocks. All the pixel values in a smooth type range block are replaced by the mean of its pixel values. The scheme mentioned above is a time saving one provided the number of smooth type range blocks is significant. The analysis of the proposed method (Section 2.6.2) discusses these aspects.

2.4.3 GA to Find Fractal Codes

The main aspect of fractal based image coding is to find a suitable domain block and a transformation for a rough type range block. Thus the whole problem can be looked upon as a search problem. Instead of a global search mechanism we have introduced GAs to find the near optimal solution.

The number of possible domain blocks to be searched are $(z-2b+1)\times(z-2b+1)$ and the number of transformations to be searched for each domain block is eight (Section 2.4.1). Thus the space to be searched consists of M elements. M is called cardinality of the search space. Here $M = 8\ (z-2b+1)^2$. Let the space to be searched be represented by $\mathcal{S}$ where

$$\mathcal{S} = \{1, 2, \cdots, (z-2b+1)\} \times \{1, 2, \cdots, (z-2b+1)\} \times \{1, 2, \cdots, 8\}.$$

Binary strings are introduced to represent the elements of $\mathcal{S}$. The set of 2^L binary strings, each of length L, are constructed in such a way that the set exhausts the whole parametric space. The value for L depends on the values of z and b. The fitness value of a string is taken to be the RMSE between the given range block and the estimated range block.

Let S be the population size and T be the maximum number of iterations for the GA. Note that the total number of strings searched up to T iterations is $S \times T$. Hence, $\frac{M}{S\,T}$ provides the search space reduction ratio for each rough type range block.

2.5 Implementation and Results

The GA based method discussed in Section 2.4 is implemented on 256×256, 8 bit/pixel "Lena" image. The image is subdivided into four 128×128 subimages, each of which is encoded separately.

For the specific implementation of the classification scheme mentioned above, the variances of pixel values of all 8×8 and 4×4 range blocks are computed and corresponding thresholds are selected from the respective histograms of the variances. For 8×8 range blocks, a valley is found near the value 20 in the histogram and thus, this value is chosen as threshold for 8×8 range blocks. Similarly, 35 is taken as the threshold value for the 4×4 range blocks. During encoding, breaking of the given image into such subimages leading towards the implementation simplicity and less computational cost. Note that the same procedure has been adopted by Jacquin [80]. Also a little gain in compression is expected as one needs to store fewer bits for domain block location.

Considering parent range blocks of size 8×8 and child range blocks of size 4×4 and using two level image partition scheme [80], each subimage is then encoded. The methodology proposed in this article is also implemented with a single level partition scheme where only parent blocks are considered.

GAs are implemented, as a search technique, only for rough type range blocks. Here for each subimage, total number of parent range blocks is $n = 256$ and total number of domain blocks (m) to search is $(128 - 16 + 1) \times (128 - 16 + 1) = 113 \times 113$ and $(128 - 8 + 1) \times (128 - 8 + 1) = 121 \times 121$ for parent and child range blocks respectively. Thus the cardinality (M) of the search space for these two cases are $113 \times 113 \times 8$ and $121 \times 121 \times 8$ respectively. The string length L has been taken to be $17(i.e., 7+7+3)$ in both the cases. As a result of selecting 2^{17} binary strings, a few strings, in both the cases, will be outside the specified search spaces. If a string which is not in the search space is selected during the implementation of the GA, then a string at the boundary will replace it.

Out of these 2^{17} binary strings, 6 strings $(S = 6)$ are selected randomly to construct an initial population. A high probability, say $P_{cross} = 0.85$ is taken for the crossover operation. For mutation operation, P_{mut} (mutation probability) is varied over the iterations and the exact values are 0.30, 0.20, 0.15, 0.10 and 0.06. Here, the mutation

probability is first varied in 5 steps starting from 0.30 to 0.06. Now, from 0.06 it goes back to 0.30. Again from 0.30 comes down to 0.06. So, altogether there are 5+4+4=13 steps in which mutation probability is varied. For each step, 70 iterations have been considered. Thus, the total number of iterations considered in the GA is $T = 13 \times 7 = 910$. Hence the search space reduction ratios for a parent and a child rough type range blocks are approximately 18 and 21 respectively.

Fig. 2.2: Original Lena image (256 × 256, 8 bpp)

For encoding "Lena" image, the results of both one level (*i.e.*, only 8×8 range blocks) and two level (*i.e.*, first 8×8 and then 4×4) encoding are reported. Test results and some statistics of both the cases are given in Table 2.1.

Table 2.1: Test results for 256 × 256, 8 bit/pixel "Lena" image

Type of encoding	Range block size	Domain block size	Number of range blocks				Compression ratio	Bits/ pixel	PSNR (in db)
			Parent		Child				
			Smooth	Rough	Smooth	Rough			
Single level	8×8	16×16	278	746	Nil	Nil	21.70	0.37	26.16
Two level	8×8 and 4×4	16×16 and 8×8	278	533	128	951	10.50	0.76	30.22

The corresponding diagrams for the "Lena" image are shown in Figures 2.2 to 2.3. Figure 2.2 shows the original "Lena" image. Figures 2.4 and 2.3 are the "decoded Lena" images (after 10 iterations) using two level and single level encodings respectively starting from the initial image given in Figure 2.5. Figure 2.6 is the result obtained in first iteration while decoding the "Lena" image starting from Figure 2.5. Figures 2.7 and 2.8 are respectively the difference images of "decoded Lena" image from the original "Lena" image after first iteration and tenth iteration.

Fig. 2.3: Decoded Lena image (0.76 bpp) using two level partitioning

Fig. 2.4: Decoded Lena image (0.37 bpp) using single level partitioning

The algorithm is also tested for the "Girl" image, the "Seagull" image and the "low flying aircraft" (LFA) image. All these images are 8 bit/pixel images of size 256×256. Figures 2.9 and 2.10 are original and decoded "Girl" images respectively. Figures 2.11 and 2.12 are original and decoded "Seagull" images respectively and Figures 2.13 and 2.14 are original and decoded "LFA" images respectively. All the decoded images are obtained after 10 iterations and with the starting image as Figure 2.5. Note that the search space reduction ratios for both these images are same as that of "Lena" image since sizes of parent and child range blocks as well as S and T are same. Test results of these images are given in Table 2.2. The detail of bit allocation for storing codes which is obtained by the proposed method is discussed in Section 2.6.2.

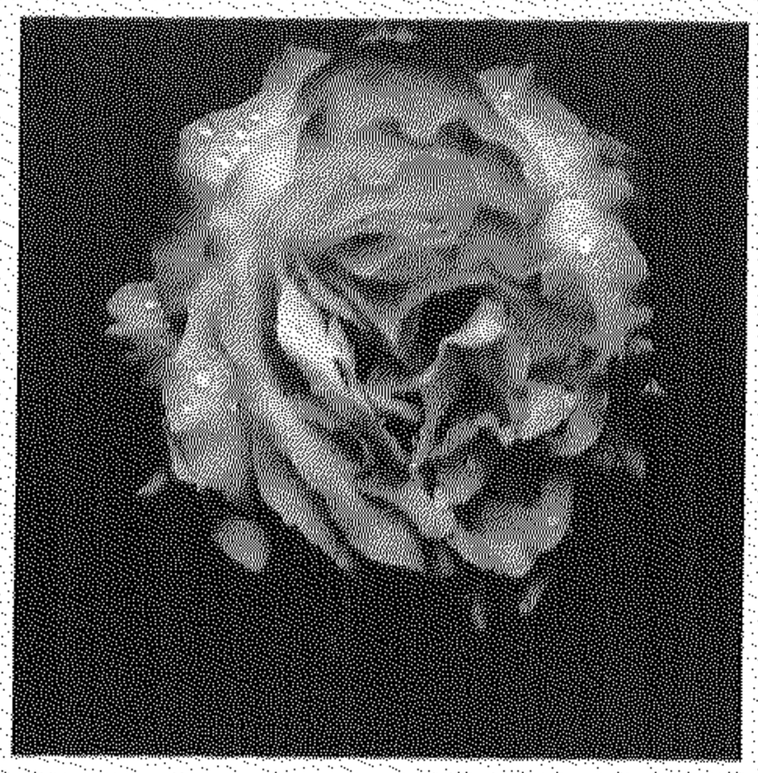


Fig. 2.5: Rose image (256 × 256, 8bpp): Initial image to start the fractal decoding

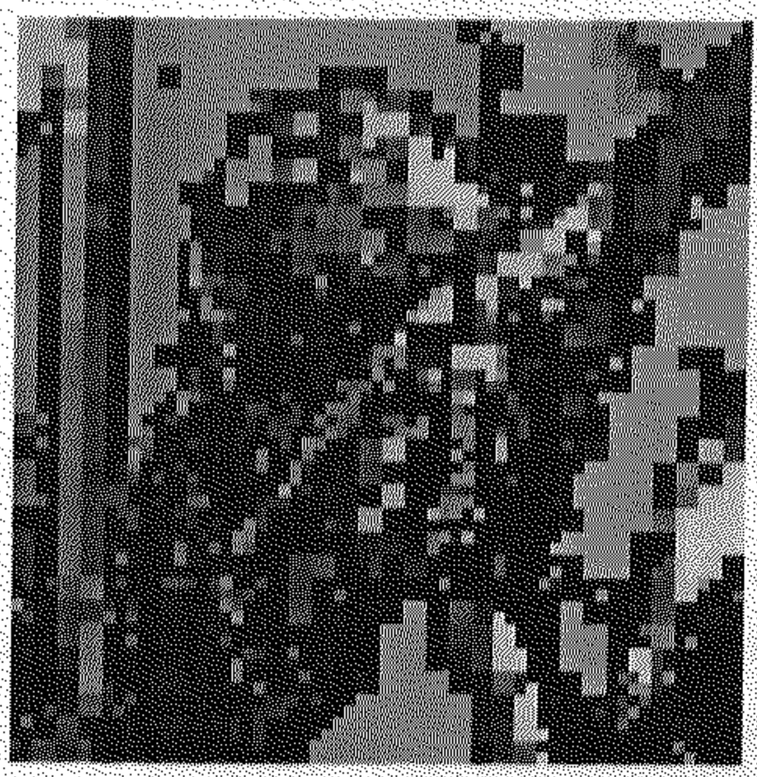


Fig. 2.6: Decoded image after first iteration while decoding Lena image

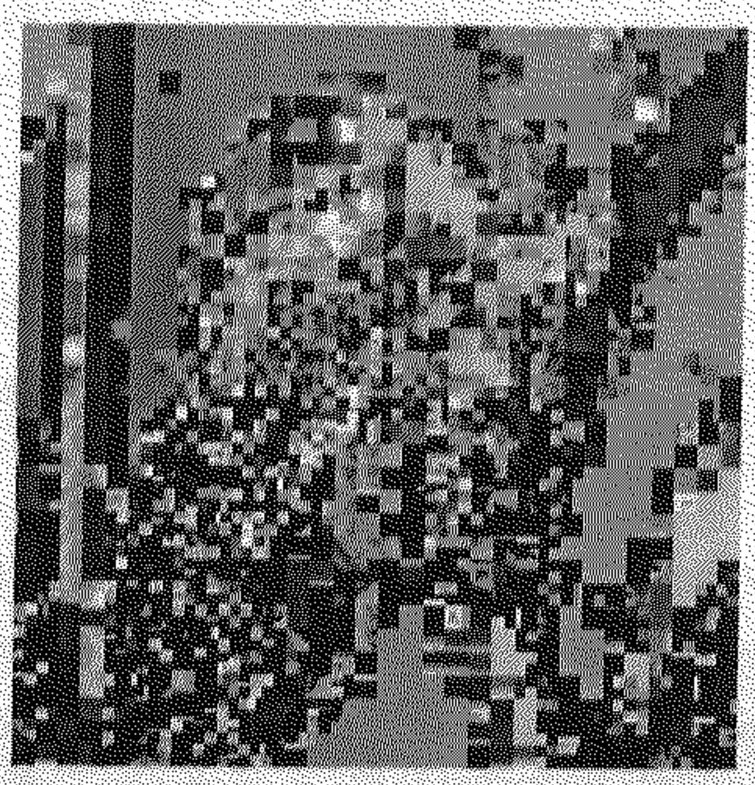


Fig. 2.7: Difference image after first iteration while decoding Lena image

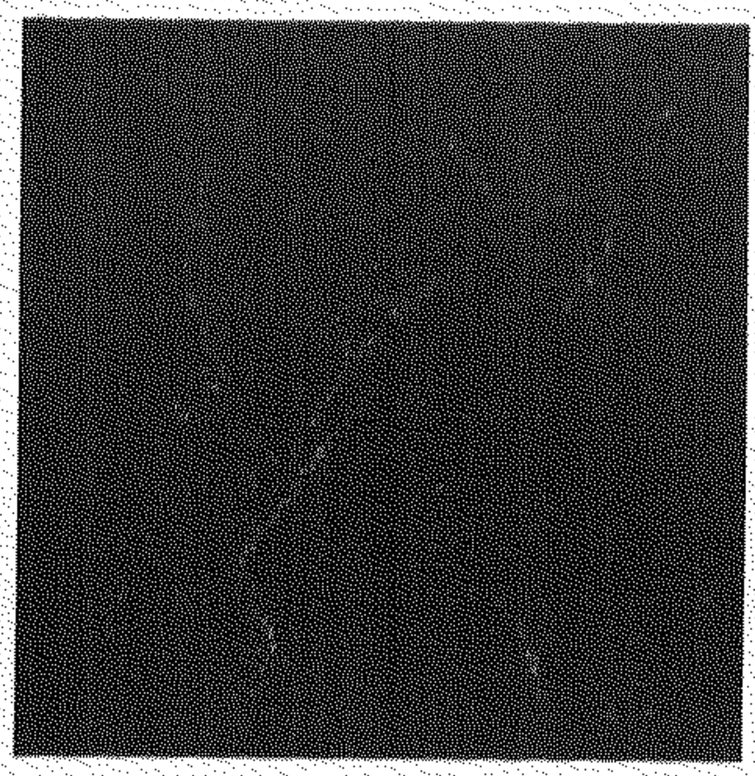


Fig. 2.8: Difference image after tenth iteration while decoding Lena image

To judge the quality of the decoded image we have used Peak-Signal-to-Noise-Ratio (PSNR). The measure PSNR is defined as follows.

If I is the original image and $\widehat{I}$ is the decoded image then

$$PSNR = -20 \log_{10} \left(\frac{\sqrt{\sum_{p} (I(p,q) - \widehat{I(p,q)})^2}}{2^t - 1} \right),$$

where, $I(p,q)$ and $\widehat{I(p,q)}$ are the pixel values corresponding to $p-th$ row and $q-th$ column of the original and the decoded image respectively. Here t is the number of bits per pixel of the image.

The GA based technique of fractal image compression method is also compared with exhaustive search mechanism. The test results of single level (8×8 range block size) and two level (8×8 and 4×4 range block size) encoding scheme of both the techniques are shown in Table 2.3. Note that all the prefixed parameters are same in both cases.

It is very clear from the test results that, for single level encoding, the compression ratios are found to be same in both techniques. On the other hand the PSNR values

Table 2.2: Test results of the GA based method for 256×256, 8 bit/pixel images

Image	Compression Ratio	Bits/ pixel	PSNR (in db)
Girl	11.37	0.70	30.74
Seagull	7.40	1.08	27.27
LFA	5.51	1.45	26.86

Fig. 2.9: Original Girl image (256×256, 8bpp)

Fig. 2.10: Decoded Girl image (0.70 bpp) using two level partitioning

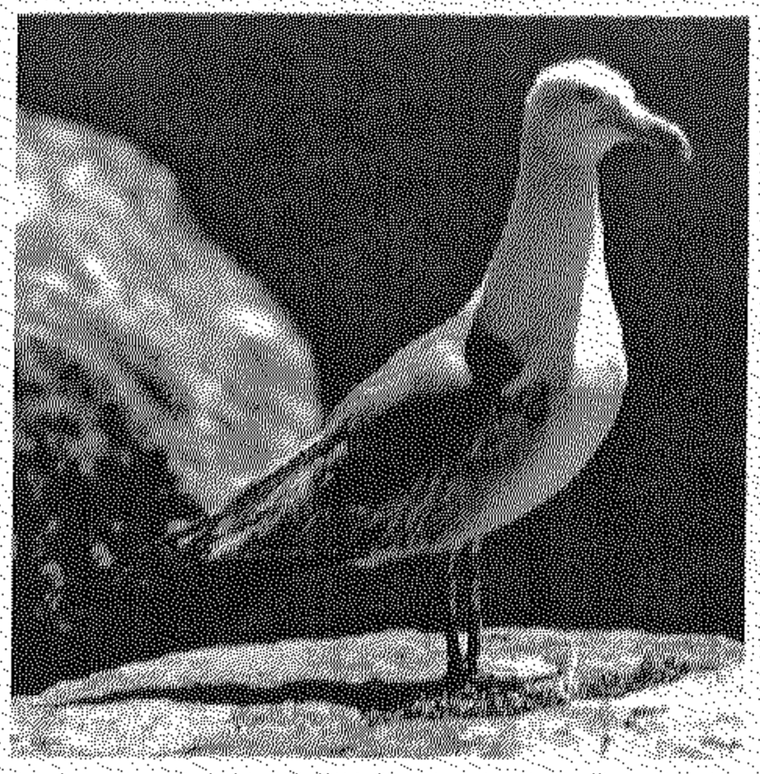


Fig. 2.11: Original Seagull image (256 × 256, 8bpp)

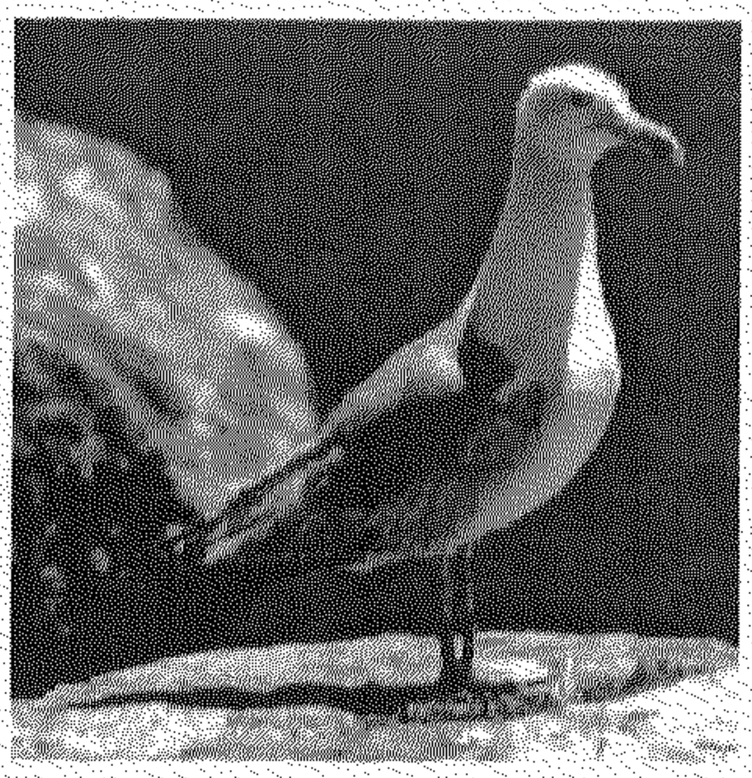


Fig. 2.12: Decoded Seagull image (1.08 bpp) using two level partitioning

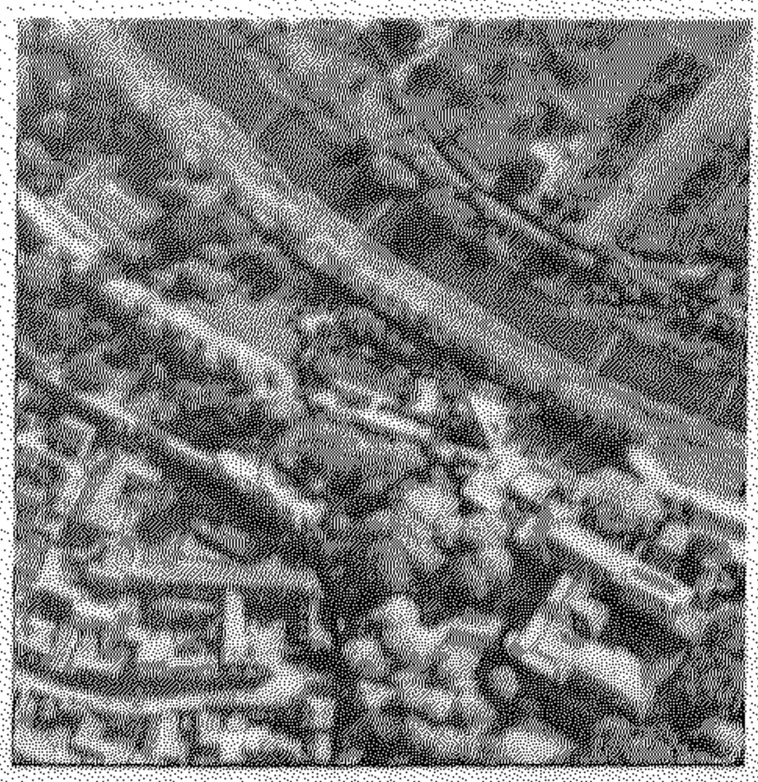


Fig. 2.13: Original LFA image (256 × 256, 8bpp)

Fig. 2.14: Decoded LFA image (1.45 bpp) using two level partitioning

of the decoded images are found to be very close to each other. But the number of domain blocks searched in both cases provides the justification of using GA as a search technique. The number of domain blocks searched in case of GA is reduced at the order of 20 than that of the exhaustive search mechanism.

In two level encoding scheme, more compression is achieved in exhaustive search. In this encoding scheme, in the first level, more rough type parent range blocks are encoded correctly (in the sense of error threshold) in exhaustive search technique. These blocks are not divided into child blocks for second level encoding. As a result of this more compression is achieved in exhaustive search case. But the PSNR value appears to be better in case of GA based technique. Moreover the advantage of using GA based technique is established from the value of the number of domain blocks searched in both the cases. Here also GA based technique searched at the order of 20 times less number of domain blocks in comparison with the exhaustive search technique.

The search space reduction is achieved since near optimal solutions are usually satisfactory and, intuitively, the solutions whose fitness values are far away from the optimal are thrown away in a bulk. This is the reason for GA to perform well for optimization problems [127].

Table 2.3: Results obtained by using the GA based technique and the Exhaustive search technique for "Lena" image

Type of encoding	Genetic Algorithm			Exhaustive Search		
	Compression Ratio	PSNR in db	Number of domain blocks searched	Compression Ratio	PSNR in db	Number of domain blocks searched
Single level 8×8	21.7	26.16	4073160	21.70	26.20	85939200
Two level 8×8 & 4×4	10.50	30.22	8102640	11.24	28.32	161869952

2.6 Comparison and Analysis

2.6.1 Comparison

The values of compression ratio and PSNR are mostly used to compare two image compression methods. Compression ratio and PSNR values are not only dependent to each other but also influenced by the other issues which are involved in coding mechanism. Usually, compression ratio decreases with the increase in the value of PSNR. The value of compression ratio is also highly influenced by the quantization procedure which is involved in storing the parameters of the coding scheme. Thus, it is desirable to bring the compression methods under the same framework for the purpose of comparison. However, it is sometimes difficult to bring a wide variety of compression schemes under the same set up. Also, different motivations may cause to give rise different compression algorithms. So, it is not always justified to compare different algorithms on the same platform. To show the acceptability of a newly suggested compression scheme among the other existing schemes, usually values of compression ratio and PSNR are reported.

The performance of a fractal image compression scheme, in terms of the encoding speed and compression, mostly depends on i) range block partitioning, ii) quantization procedure and iii) initial domain pool (search space) used. Let us discuss how these influence the performance of the GA based fractal compression. The main advantage of the usage of GA is to speed up the encoding process to find the best matched domain blocks and transformations for range blocks. This advantage has been shown by comparing the results of the present method with that of the compression scheme where GA has not been considered. Same non overlapping square range block partitioning and the same quantization procedure has been used in both the schemes (see Table 2.3).

The range block partitioning plays an important role in fractal compression. In order to verify the efficiency of GA in case of searching, it makes more sense to apply the basic philosophy of GA, considering a fixed quantization procedure, in different fractal compression methods where different range block partitioning are used. In this context, two square block partitioning schemes, *viz.*, single level non overlapping and two level non overlapping have been considered. The results obtained by the present GA based

method using two level partitioning appears to be better than the results of the single level partitioning. In this regard, one can think of using other partitioning schemes such as quadtree, HV, triangular and irregular. This may lead to other efficient fractal image compression schemes. In the present investigation, our aim is not only restricted to examine the usage of GA but to suggest a fractal image compression algorithm which is fast as well as efficient in the sense of compression ratio and quality of the decoded image.

The quantization process influences the compression ratio and there by the quality of the decoded image. The specific quantization process needs to be explained in case of comparing several compression algorithms. The usage of GA, as in our investigation, is not directly influenced by the quantization. No quantization has been used to store the parameters *viz.*, location of the domain block and isometric transformation which are found out using GA.

Initial domain pool or the search space, used for finding the matched domain block of a range block, mainly influences the speed of the encoding process. Classification is a widely used scheme for trimming of total search space. The process of finding matched domain block from a trimmed domain pool may lead to poor quality of the decoded image. GA can be adopted in case of searching from a sub set of the total space. However, it is always better to utilize the efficiency of GA as a search mechanism on a large search space. With this in mind no restriction has been imposed in order to partition the initial domain pool.

The implementation details of a few existing fractal image compression methods along with the present method is discussed here. This discussion takes care of two aspects *viz.*, i) how the present scheme differs from the existing schemes and ii) how the basic notion of GA can be adopted within existing schemes.

A different kind of trimming of the search space was described by Jacquin [80] to reduce the search space. In his method the reduction takes place in two steps. 1) A "domain pool" consisting of some "domain blocks" is constructed. 2) The domain blocks in the domain pool are classified on the basis of some geometric features.

In Jacquin's method starting from the first pixel of the given image, domain blocks are selected by sliding a window of size equal to the size of the domain block across the image and taking a constant shift horizontally and vertically. Selecting the domain

pool consisting of domain blocks of sizes 16×16 and 8×8, shifts of 4 pixels and 2 pixels respectively are considered. Thus the reduction ratios are 16 and 4 for 8×8 and 4×4 range blocks respectively for a 256×256 image. On the other hand, the GA based method reduces the search space corresponding to domain blocks and isometric transformations simultaneously and the search space reduction ratios for a parent and a child rough type range blocks are approximately 18 and 21 respectively. Moreover, the best matched domain block corresponding to a range block can be located any where within the image support, and so, on trimming the maximal domain pool by shift method, we may lose the best matched domain block in the Jacquin's method. But the GA based method utilizes the maximal domain pool while searching for the best matched domain block.

The second part of the reduction is obtained using the classification scheme proposed by Ramamurthi and Gersho [146]. This three class classification scheme is adopted to classify the pool of range blocks as well as the pool of domain blocks. This scheme has advantages both in efficient encoding of the range blocks as well as in reducing its search space. But it has limitations too. According to the scheme, both pools are classified into three classes, so two bits may be required for storing to indicate the class of the range blocks under consideration. In the present encoding algorithm, the range blocks only are classified into two classes which is not only a time saving scheme but it also requires storing of only one bit for class information thereby providing more compression.

The compression ratio of a fractal compression scheme depends on the number of range blocks used. More compression can be achieved by considering less number of range blocks. Considering fewer number of range blocks by using quadtree method and HV partitioning method, Fisher et al. [58] designed a scheme which results in an increase of compression ratio. The compression ratio (9.97) and the PSNR (31.53) reported by Fisher et al. [58] are almost equal to those reported in Section 2.5.

The search space in the methodology described by Fisher et al.[58] is dependent on the complexity of the given image. In particular, in the first step of quadtree method, best domain block for only four range blocks has to be searched. The search process is then carried out up to a fixed level where the minimum size of the range blocks is fixed. The search, in all the intermediate steps, has to be done exhaustively to reach the fixed lowest level in the quadtree method. Thus, an extensive search may need to

be carried out for some images. So, in comparison with this method, the proposed GA based method is better for reducing the search space. Note that, the proposed GA based (which is a two level scheme) can be extended to a quadtree scheme containing multiple levels, with or without classification of range blocks.

The basic philosophy of the proposed GA based technique can be adopted to other fractal based coding techniques too. Thomas et al.[170] suggested an algorithm for fractal based image compression in which the neighborhood information plays an important role in increasing the compression ratio. In that method, the domain block for a "seed" range block is found by an exhaustive search mechanism. The domain blocks for the other range blocks are found by utilizing the connectivity of the range blocks. One can adopt the proposed GA based technique for finding the domain block for the "seed" range block.

The main contribution of this chapter involves in developing a fractal image compression algorithm which is fast as well as efficient in the sense of compression and quality of the decoded image. GA has been used to speed up the encoding process and for the specific implementation two range block partitioning schemes have been used. Now finally, to show the acceptability of the result obtained by the present scheme, test results of some existing fractal compression techniques on "Lena" image, as reported, are presented in Table 2.4. The brief description of the existing methodologies, for finding PIFS code of an image, is provided in Chapter 1 Section 1.3. The present GA based algorithm has also been implemented on a 512 × 512, 8 bits/pixel "Lena" image using single level and two level encoding schemes. In single level encoding, range blocks of size 16 × 16 have been considered. On the other hand, in a two level partitioning scheme, range blocks of sizes 16 × 16 (parent) and 8 × 8 (child) have been considered. The compression ratio and PSNR values are found to be 77.25 and 26.13 respectively for single level scheme and 37.16 and 30.87 respectively for two level scheme. A summary of the test results on "Lena" image is shown in Table 2.5. Original "Lena" image of size 512 × 512 as shown in Figure 2.15 is coded. The result of the decoded image of a two level scheme is shown in Figure 2.16 and the same for the single level scheme is shown in Figure 2.17.

It has been observed from Table 2.4 and Table 2.5 that test results of the proposed GA based fractal image compression scheme are satisfactory.

Fig. 2.15: Original Lena image (512 × 512, 8bpp)

Fig. 2.16: Decoded Lena image (512 × 512, 8bpp) using two level partitioning

Fig. 2.17: Decoded Lena image (512×512, 8bpp) using single level partitioning

Table 2.4: Test results of some fractal image compression schemes on "Lena" image

Input image	Article	Compression ratio	PSNR (in db)
256 × 256,6bpp	[80]	8.80	27.70
256 × 256,8bpp	[58]	9.97	31.53
256 × 256,8bpp	[150]	16.00	29.10
256 × 256,8bpp	[69]	10.60	30.72
512 × 512,8bpp	[99]	40.00	30.20
512 × 512,8bpp	[100]	44.00	29.10
512 × 512,8bpp	[170]	41.00	26.56
512 × 512,8bpp	[151]	53.30	30.30
512 × 512,8bpp	[169]	44.44	29.10
512 × 512,8bpp	[58]	36.78	30.71
512 × 512,8bpp	[152]	69.50	28.30
512 × 512,8bpp	[41]	65.60	29.90

Table 2.5: Test results of the proposed GA based compression schemes on "Lena" image

Input image	Type of encoding	Compression ratio	PSNR (in db)
256 × 256,8bpp	Single level	21.70	26.16
256 × 256,8bpp	Two level	10.50	30.22
512 × 512,8bpp	Single level	77.25	26.13
512 × 512,8bpp	Two level	37.16	30.87

2.6.2 Analysis

Before going to analyze the method, let us first describe the detail of bit allocation for storing the fractal code of the present algorithm. The fractal code of a smooth type range block consists of only one parameter *viz.*, the mean gray level value. For a 256 × 256, 8 bits per pixel image, this parameter (integer value) requires 8 bits for storage. On the other hand, fractal code of a rough type range block consists of five parameters. The first two indicate the location of the matched domain block. These are, in particular, starting row and start column of the matched domain block. These two are again integers. As four subimages of the given 256 × 256, 8 bits per pixel image are encoded separately, 7+7=14 bits are required to store these two parameters. The next parameter is the isometric transformation used. Only 3 bits are required to store a transformation out of 8 isometric transformations. Last two are scale and shift parameters to find the estimated values of the range block from the matched domain block. The scale parameter is quantized into 4 values *viz.*, 0.7, 0.8, 0.9 and 1.0. Thus only 2 bits are required for the storage of this parameter. Now, for the choices of the scale parameter, the value of the shift parameter is ranging from -255 to +255. We have considered only integer values to quantized this parameter. Hence 9 bits are required to store this shift parameter. Out of this 9 bits, the first bit indicates the sign (+ or -) and the rest of the bits indicate an integer value between 0 and 255. Thus 7+7+3+2+9=28 bits are required for the storage of the fractal code of a rough type range block.

Moreover a two level image partition scheme (parent and its four children) as implemented takes care of finer details of a very small portion of the image. There are altogether 12 possible configurations of four children along with their parents. To indicate the location and presence of child blocks of a parent block, four extra bits for each parent block are needed during encoding. Likewise an extra bit for each transformation (parent and child) is needed if classification scheme, as mentioned here, is adopted. Thus, the validity of using the classification scheme together with the two level image partition scheme need to be investigated. Comparison of performances of different encoding schemes (single level, two level, without classification, with classification) can be made from the point of view of compression and quality of the decoded image. Table 2.6 shows the number of bits required for a 256 × 256, 8 bit/pixel image under different situations in different schemes. Number of range blocks are 1024 and 4096 with sizes 8×8 and 4×4 respectively. Number of bits for storing each transforma-

tion is 28 in the coding scheme without using any classification scheme. In the coding scheme using classification, 9 bits and 29 bits respectively are necessary for storing the transformation for each smooth type and rough type range blocks.

Table 2.6: Number of bits necessary for storing transformation in different schemes of a 256×256, 8 bpp image.

		Single level (8×8)	Two level ((8×8) & (4×4))		
			All parent	All child	Mixed (parent & child)
Without classification		28×1024	$28 \times 1024 + 1024 \times 4$	$28 \times 4096 + 1024 \times 4$	$28 \times c + 1024 \times 4$
With classification	All smooth	9×1024	$9 \times 1024 + 1024 \times 4$	$9 \times 4096 + 1024 \times 4$	$9 \times c + 1024 \times 4$
	All rough	29×1024	$29 \times 1024 + 1024 \times 4$	$29 \times 4096 + 1024 \times 4$	$29 \times c + 1024 \times 4$
	Mixed (smooth & rough)	$29 \times 1024 - 20 \times r$	$29 \times 1024 - 20 \times r + 1024 \times 4$	$29 \times 4096 - 20 \times r + 1024 \times 4$	$29 \times c - 20 \times r + 1024 \times 4$

Table 2.6 shows the number of bits necessary for storing transformations in different coding schemes for a 256×256, 8 bpp image. The selection of a particular scheme for a given image, which provides suitable values for the PSNR and the compression ratio, depends upon the values of "c" and "r". Here "c" is the total number of codes in two level partitioning where both parent and child blocks are present and "r" is the number of smooth type range blocks under classification scheme. The following conclusions regarding the usefulness of the application of different schemes during encoding process may be extracted from Table 2.6.

- If a "single level" scheme provides the desired quality of the decoded image (*i.e.*, the desired PSNR) then one should not opt for a "two level" scheme. Note that a two level scheme decreases the compression ratio.
- If all the blocks in a "two level" scheme are of same size then "single level"

encoding with that block size is preferable.

- Under no classification, a "two level" encoding scheme where both parent and child blocks are present is preferable to"single level" scheme if $c < 877$. On the other hand, "single level" encoding with block size 4×4 is preferable to this scheme when $c > 3950$. If c lies between 877 and 3950 then switching over from "single level" to "two level" depends upon the quality of the decoded image, (*i.e.*, PSNR).

  Under classification, when all the blocks present are "smooth" type, a "two level" encoding scheme where both parent and child blocks are present is preferable to "single level" scheme if $c < 568$. Unlike this, "single level" encoding with block size 4×4 is preferable to this scheme when $c > 3641$. If c lies between 568 and 3641 then switching over from "single level" to "two level" depends upon the quality of the decoded image, (*i.e.*, PSNR).

  Similarly, with classification, when all the blocks present are "rough" type or "mixed" type (both smooth and rough type blocks are present), a "two level" encoding scheme where both parent and child blocks are present is preferable to "single level" scheme if $c < 880$. On the other hand, "single level" encoding with block size 4×4 is preferable to this scheme when $c > 3955$. If c lies between 880 and 3955 then switching over from "single level" to "two level" depends upon the quality of the decoded image, (*i.e.*, PSNR).

- Let r_1 be the number of parent smooth blocks out of c number of codes. Let $29 \times c + 60 \times r_1 > 28 \times 4096$ and also let every child block of every smooth parent block be a smooth block. [Note that there are four 4×4 child blocks of each 8×8 parent block.] Then "single level with classification" scheme with block size 4×4 is preferable to "two level with classification" scheme.

- If all the blocks belong to the same class then "classification" is not needed.

- If both smooth and rough type blocks are present (after classification) with number of smooth blocks less than 5% of the total codes present then classification is not needed.

Note that similar inferences may be drawn for the other available fractal based image coding schemes.

2.7 Implementation of GA Based Method on Colour Images

A common format for images is 24 bits/pixel colour. A colour image can be split into three different colour components, *e.g.*, red, green and blue. Each component can be treated as an independent gray scale image and can be represented by 8 bits/pixel. As a relatively large amount of information is used to store each pixel, there is a large potential for compression of such images. The compression of colour images can be achieved by compressing three separate gray scale images. However the potential for compression is enhanced by the fact that there is often a high degree of correlation between the pixel values of the three bands, red, green and blue, of images. In fact, colour redundancies can be exploited to get high compression. Possible applications of fractal technique in colour images using suitable decomposition of colour components [106] and vector distortion measure [181] are available in the literature.

There are several transformations to decompose the prime colour (RGB) components to different independent components causing the removal of some colour redundancies. Different decompositions decorrelate the colour redundancies differently. As a consequence, the performance of a compression system depends on the colour image decomposition. Decomposition of a colour image into one luminance (Y) image and two chrominance (I and Q) images is the method that has been used in most commercial applications including colour television system. The luminance (Y) image generally contains the bulk of the information content of the colour image. The chrominance (I and Q) images contain the information corresponding to the image colour content, a small fraction of the image information. The conversion of primary RGB components to YIQ components and the converse are as follows:

$$\begin{pmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{pmatrix} \begin{pmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{pmatrix} = \begin{pmatrix} \mathrm{Y} \\ \mathrm{I} \\ \mathrm{Q} \end{pmatrix}$$

$$\begin{pmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.273 & -0.647 \\ 1.000 & -1.104 & 1.701 \end{pmatrix} \begin{pmatrix} \mathrm{Y} \\ \mathrm{I} \\ \mathrm{Q} \end{pmatrix} = \begin{pmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{pmatrix}$$

The proposed GA based method of fractal image compression scheme has been applied separately on Y image, I image and Q image. Note that these three images are 8 bits/pixel gray scale images. During decoding, approximations of Y, I and Q are obtained and the final decoded colour image is obtained by transforming approximated YIQ to RGB. But before applying the GA based fractal method on Y, I and Q, two pre-compression schemes have adopted. The first pre-compression scheme consists of reducing the pixel information from 8 bits to 6 bits in such a way that the loss of getting information of all 8 bits from 6 bits, in a decompression procedure, is minimum. This compression-decompression (*i.e.*, 8 bits to 6 bits and again 6 bits to 8 bits) is called lift-down and lift-up scheme and this has been applied to the luminance (Y) image only.

Lift-up and lift-down schemes are used in digital information hiding [84]. The information to be hidden is packed in the least significant bit(s) of an image resulting a little loss in quality of the image. In the present case, in lift-down scheme the last two insignificant bits are purposefully deleted to get only 6 bits. These 6 bits represent the same image. On the other hand the lift-up scheme consists of randomly generating two bits and adding those as least two insignificant bits with the existing 6 bits. The new 8 bits represent the same image too. Note that a little loss in quality of the decoded image is expected on using lift-down and lift-up schemes. But a little gain in compression is also expected. Thus one needs to make a trade-off in between quality and compression. these two schemes have not been adopted in case of gray level images so as to get the quality decoded images.

The second pre-compression scheme allows for spatial averaging of chrominance images with little loss in image fidelity as chrominance images contain only a small fraction of the image information. In the decoding scheme, the fractal decompression of YIQ are followed by lift-up of Y image and spatial expansion of I and Q images. A block diagram of the whole process is given in Figure 2.18.

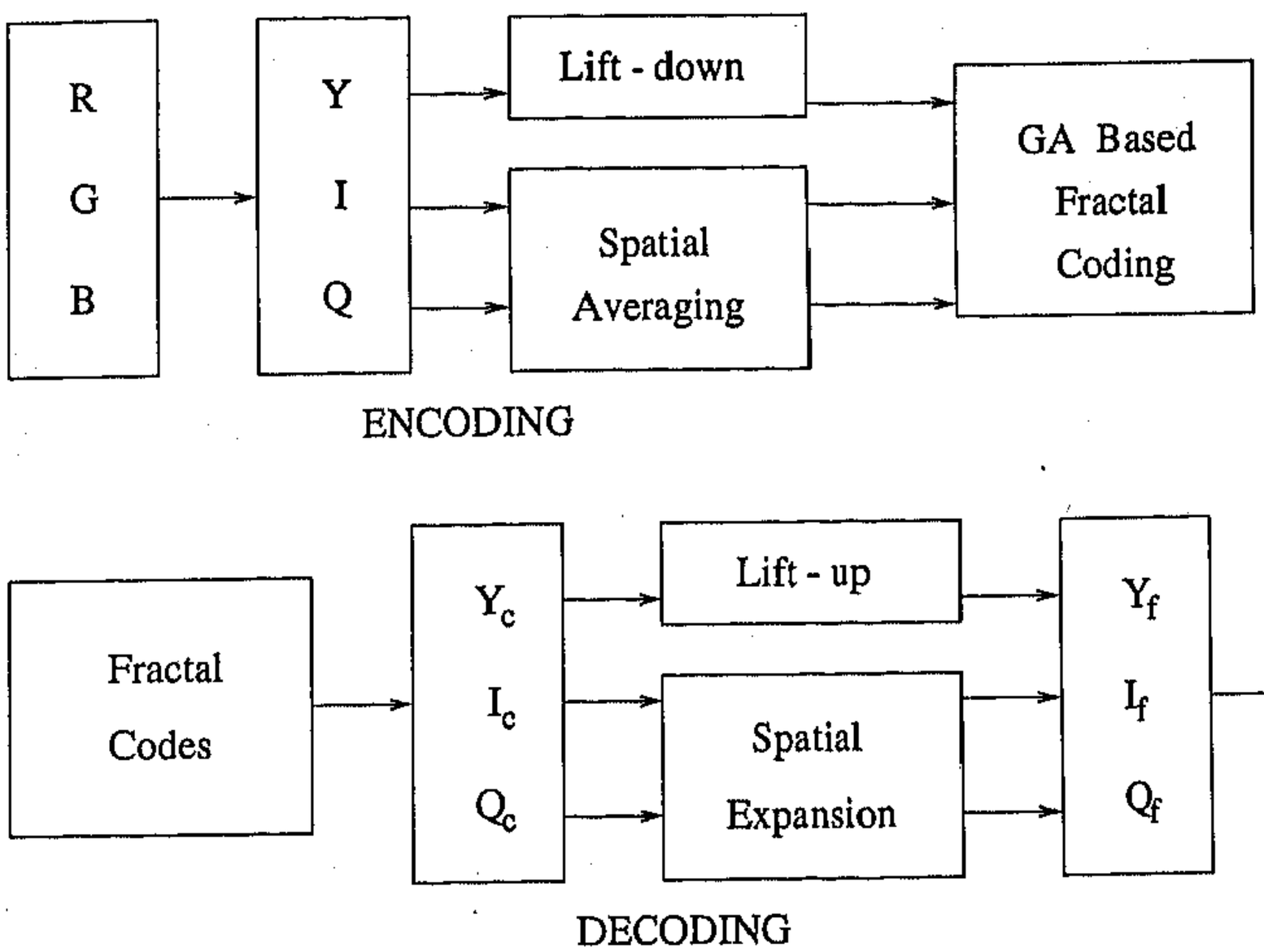
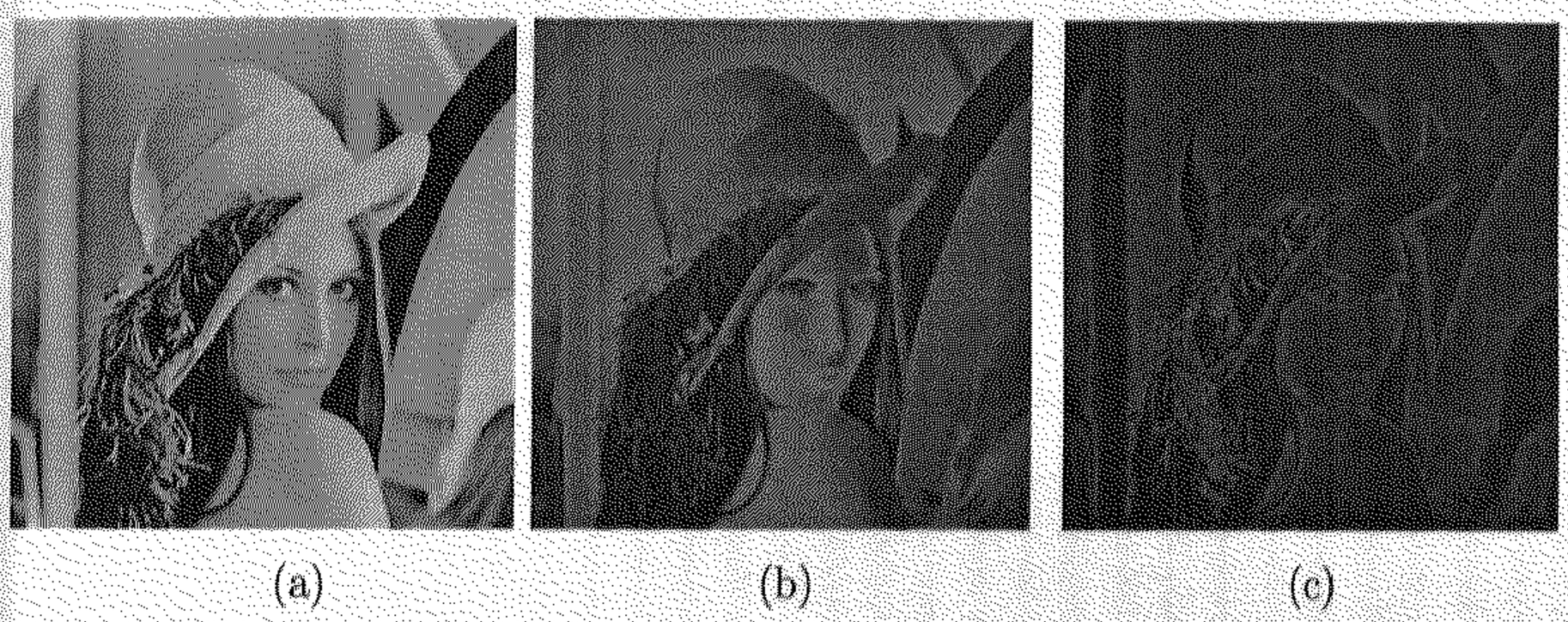


Fig. 2.18: Block diagram of the proposed GA based fractal image compression technique for colour images

2.7.1 Results

The proposed method has been implemented on several 24 bits/pixel colour images of size 256 × 256. RGB components are first converted to YIQ components. Figure 2.19 shows luminance image and chrominance images of "Lean" formed from the colour (RGB) components. Pre-compression schemes are applied before adopting the GA based method. The spatial averaging of I and Q has also been carried out. Test results and some statistics are shown in Table 2.7.



(a) (b) (c)

Fig. 2.19: Luminance and chrominance images of Lena : (a) is Y, (b) is I and (c) is Q

Figures 2.20, 2.22 and 2.24 are respectively the original images of "Lena", "Peppers" and "Fig" images. Corresponding decoded images are shown in Figures 2.21, 2.23 and 2.25.

The proposed method has also been tested in a 512 × 512, 24bits/pixel "Lena" image. Figures 2.26 and 2.27 are respectively the original and decoded "Lena" image. In this case the compression ratio and PSNR are found to be 85.46 (0.28 bpp) and 27.08 respectively.

ble 2.7: Test results of GA based method for 256 × 256, 24 bit/pixel colour images

Image	Number of range blocks				Compression Ratio	Bits/ pixel	PSNR (in db)
	Parent		Child				
	Smooth	Rough	Smooth	Rough			
Lena					16.64	1.44	27.52
Y	42	656	525	1321			
I	5	116	123	506			
Q	38	147	72	316			
Peppers					18.30	1.29	28.62
Y	166	554	495	1175			
I	11	159	61	365			
Q	43	156	40	283			
Fig					19.03	1.26	27.80
Y	160	559	535	1184			
I	7	141	129	482			
Q	42	198	63	96			

ig. 2.20: Original colour Lena image (256 256, 24bpp)

Fig. 2.21: Decoded colour Lena image (1.44 bpp)

Fig. 2.22: Original colour Peppers image (256 × 256, 24bpp)

Fig. 2.23: Decoded colour Peppers image (1.29 bpp)

Fig. 2.24: Original colour Fig image (256 x 256, 24bpp)

Fig. 2.25: Decoded colour Fig image (1.26 bpp)

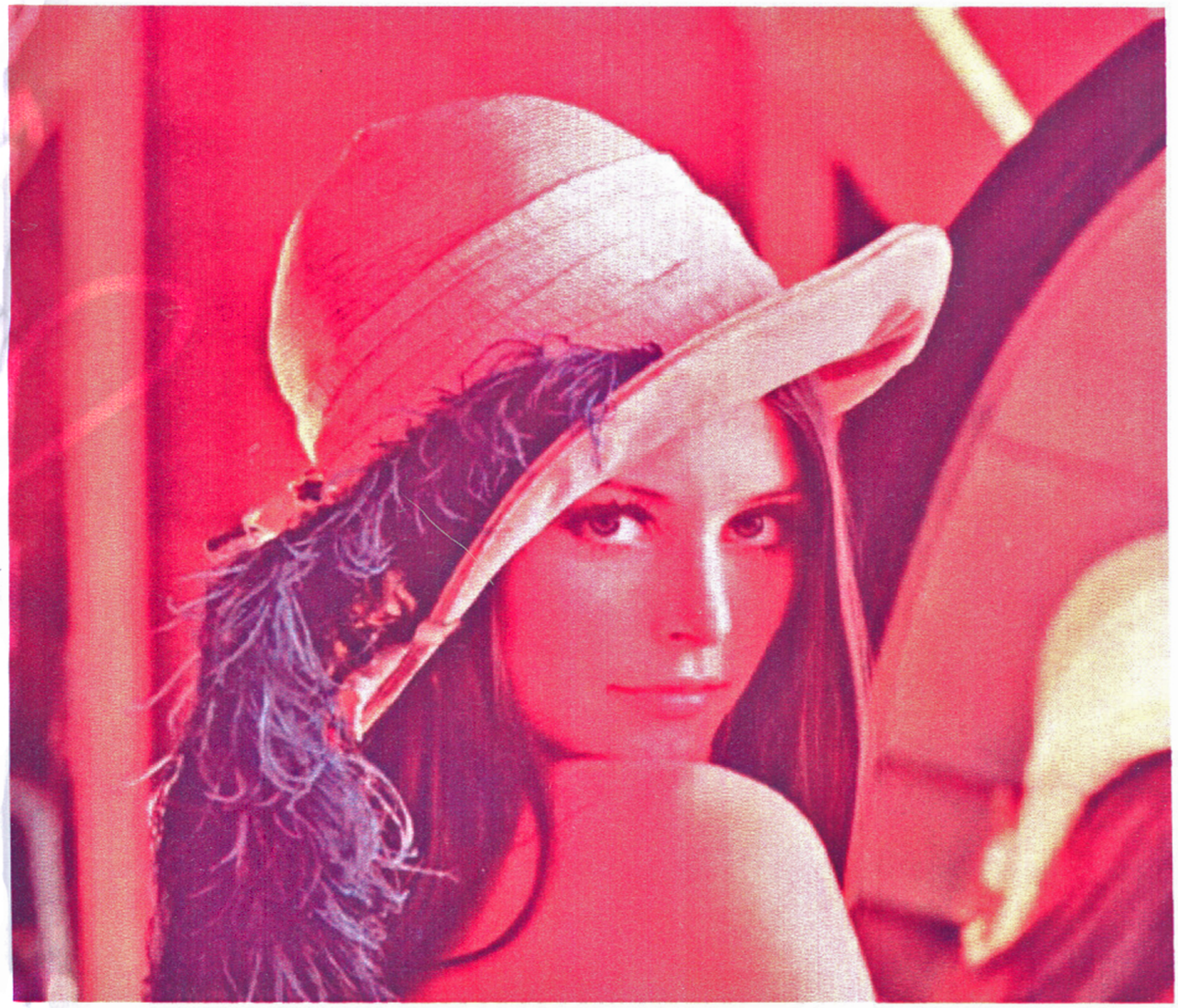


Fig. 2.26: Original colour Lena image (512 × 512, 24bpp)

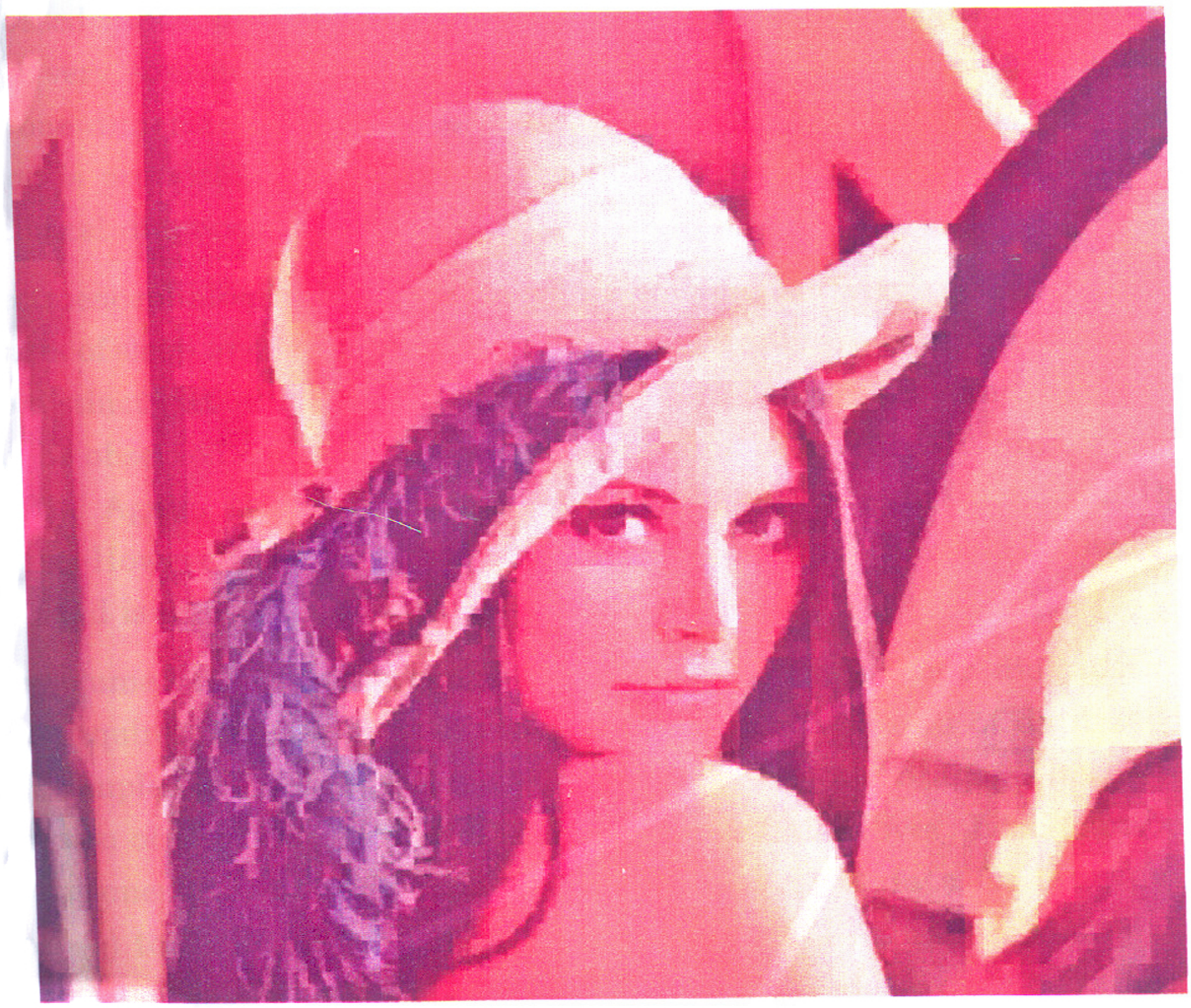


Fig. 2.27: Decoded colour Lena image (0.28 bpp)

2.8 Implementation of GA Based Method on One Dimensional Signal

This section deals with the extension of the GA based fractal coding technique to the case of one dimensional signals such as EEG signals. The signals like EEG, ECG and earthquake are seemingly very irregular in nature. The objective here is to develop a fractal based methodology for encoding one dimensional signals whose variability over time is very complex. The use of fractal technique for modeling [182] and coding [107, 131] one dimensional signals has not been explored much. A method for EEG compression is proposed here keeping the quality of the decoded signals sufficiently good for clinical diagnosis.

EEG (Electroencephalography) reflects the electrical activity of the brain during the various states of sleep and wakefulness. The EEG signals are complex in nature. During sleep, two distinct patterns, *viz.*, REM (Rapid Eye Movement) and SWS (Slow Wave Sleep) can be observed in EEG. These two patterns are easily distinguishable visually (Figure 2.28). Again, the SWS recordings of EEG is quite distinct from those of the wakeful state. However, the REM sleep recordings resemble the wakeful state recordings very much, and is difficult to identify by EEG alone [47].

The sheer volume of EEG recordings is not easy to handle for diagnostic purposes. Moreover it requires huge amount of storage space. Each second of digitized EEG data (2 channels sampled at 256 Hz) takes a kilobyte of space for storage. Therefore, the need for compression of the data is of utmost importance. Moreover, EEG compression can help (*a*) to augment the storage capacity of collected EEG data for later evaluation or comparison, (*b*) to facilitate transmitting real-time EEG signals to distant places and (*c*) to transmit rapidly and economically off-line EEG data over telecommunication networks to remote interpretation centers. Unfortunately, enough importance has not been paid towards solutions of these problems. We have came across only a few articles [19, 70, 85, 143, 144, 171] in this regard.

The method proposed for image compression is suitably modified here for one dimensional signal compression. Out of several possible transformations, eight transformations have been selected and the search parameters have been redefined for the implementation of the algorithm. As before, the basic idea of this technique is to approximate

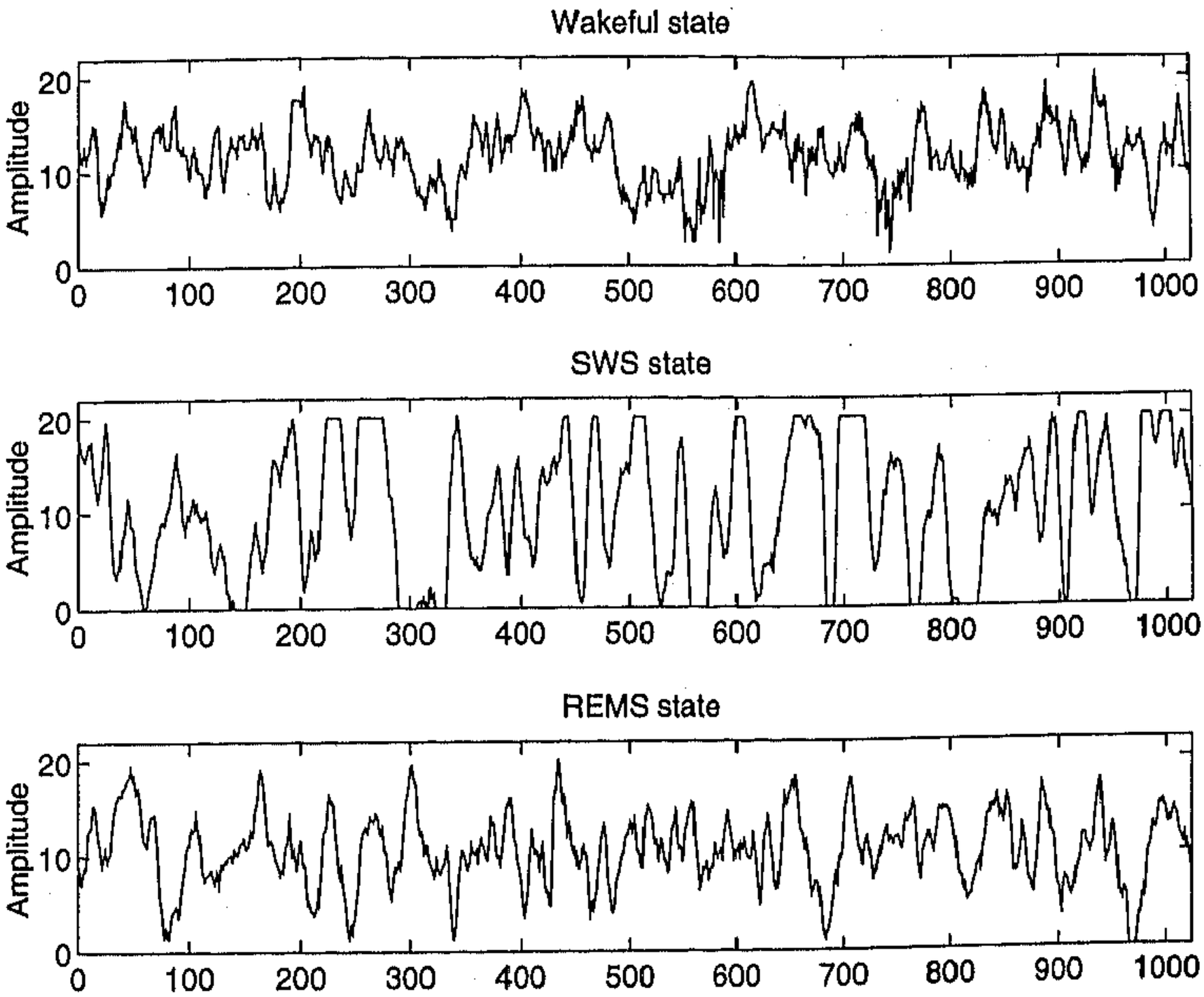


Fig. 2.28: Original EEG signals in three states

the given signal using a set of affine contractive transformations. The set of affine contractive transformations, through iterative process, produces a signal which is very close to the target signal. The main task, as in the case of image compression, is to find the appropriate set of transformations whose attractor approximates the given signal. The aforesaid transformations are obtained using the similarities present in the given signal. Here similarity implies that the waveform of a particular segment, called the range segment, of the signal is a scaled and transformed version of another segment, called domain segment, of the same signal. As in the case of image compression, Genetic Algorithms, are utilized to decreases the search space for finding an appropriate transformation and location of the matching domain segment for a range segment of the given signal. The fidelity of the reconstructed signal obtained by the present compression algorithm has been assessed. The compression ratios for the EEG signals are found to be comparable to those obtained from the other available techniques for EEG compression.

2.8.1 Methodology for Encoding EEG Signal

In this section, initially, the procedure of collecting EEG data used in this chapter is described. Then the encoding and decoding mechanisms are described.

2.8.1.1 Electroencephalogram

The EEG data has been collected from male rats (*Rattus norvegicus var. albinus*) of Charles Foster strains. The EEG has been recorded on paper through an 8-channel polygraph and sampled at 256 Hz through a 12-bit analog- digital converter into the hard-disk [153].

2.8.1.2 Generation of Fractal Codes using Affine Transformations

The process of construction of fractal codes for EEG signals is almost same as that of an image with an exception regarding the dimension of the defined contractive transformations. Let V be a given signal having z points with a range of amplitude values $\{1, 2, \cdots, l\}$. Thus the given signal V is a subset of $\mathbb{R}^2$. The signal is first partitioned

into n non overlapping segments having say b number of points, and let this partition be represented by $\mathcal{N} = \{\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_n\}$. Each $\mathcal{R}_i$ is called as a range segment. Note that $n = \frac{z}{b}$. Let $\mathcal{M}$ be the collection of all possible segments having $2b$ number of points. Let $\mathcal{M} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m\}$. Each $\mathcal{D}_j$ is called a domain segment with $m = (z-2b+1)$.

Let,

$$\mathcal{G}_j = \{g : \mathcal{D}_j \to \mathbb{R}^2 \; ; \;\; g \;\text{ is an affine contractive map}\}.$$

Now, for a given range segment $\mathcal{R}_i$, let, $g_{i|j} \in \mathcal{G}_j$ be such that

$$\delta(\mathcal{R}_i,\ g_{i|j}(\mathcal{D}_j)) \;\leq\; \delta(\mathcal{R}_i,\ g(\mathcal{D}_j)) \;\; \forall g \in \mathcal{G}_j,\ \forall j.$$

Here "δ" measures distance between two sets of points.

Now let k be such that

$$\delta(\mathcal{R}_i,\ g_{i|k}(\mathcal{D}_k)) \;=\; \min_j\{\ \delta(\mathcal{R}_i,\ g_{i|j}(\mathcal{D}_j))\ \} \tag{2.5}$$

Also, let $g_{i|k}(\mathcal{D}_k) = \widehat{\mathcal{R}}_{i|k}$. Our aim is to find $g_{i|k}(\mathcal{D}_k)$ for each $i \in \{1, 2, \cdots, n\}$. In other words, for every range segment $\mathcal{R}_i$, one needs to find an appropriately matched domain segment $\mathcal{D}_k$ as well as an appropriate transformation $g_{i|k}$.

To find the best matched domain segment as well as the best matched transformation, we are to search all possible domain segments as well as all possible transformations with the help of equation (2.5). The affine contractive transformation $g_{i|k}$ is constructed using the fact that the points of the range segment are scaled and shifted versions of the points of domain segment. Thus the affine transformation has two parts. The first part indicates which point of the range segment corresponds to which point of domain segment. The second part is to find the scaling and shift parameters.

The first part is shuffling the points of the domain segment and can be achieved by using any one of the eight possible transformations on the domain segments as described below in Section 2.8.1.3. Once the first part is obtained, second part is estimation of a set of values (amplitude) of range segments from the set of values of the transformed domain segments. These estimates can be obtained by using the least square analysis of two sets of values.

The distance measure "δ" [used in equation 2.5] is taken to be the simple root mean square error (RMSE) between the original set of amplitude values and the obtained

set of amplitude values of the concerned range segment. Let $\mathcal{R}_i(p)$ and $\widehat{\mathcal{R}}_{i|k}(p)$ be respectively the original and the obtained values of the p^{th} point of the range segment $\mathcal{R}_i$. Thus, the expression for RMSE will be

$$\delta(\mathcal{R}_i,\ \widehat{\mathcal{R}}_{i|k}) \ = \ \frac{1}{b}\sqrt{\sum_{p=1}^{b}\ (\mathcal{R}_i(p) - \widehat{\mathcal{R}}_i(p))^2}. \tag{2.6}$$

As the selection of fractal code for a range segment is dependent only on the estimation of amplitude values of that segment, it is enough to calculate only the distortion of the original and estimated amplitude values of the segment. Thus, RMSE is taken as the distance measure.

2.8.1.3 Class of Transformations

We have stated that a contractive affine transformation $g_{i|j}$ defined on $\mathbb{R}^2$ is such that $g_{i|j}(\mathcal{D}_j) \rightarrow \mathcal{R}_i$. Also $g_{i|j}$ consists of two parts, one for spatial information and the other for information of amplitude values. The second part is obtained using least square analysis of two sets of points once the first part is fixed. Moreover the size of the domain segment is double that of the range segment. But, the least square (straight line fitting) techniques needs point to point correspondence. To overcome this, the amplitude values of a range segment correspond to the average values of two consecutive points in the domain segment, thus making contracted domain segment correspond to the range segment. Note that the size of the contracted domain segment is equal to the size of the range segment.

The first part of the transformation $g_{i|j}$ indicates which point of the contracted domain segment correspond to which point of the range segment. The following eight transformations simply shuffle the points within a contracted domain segment so that it can correspond to the range segment in eight different ways.

Note that we have considered the size of the range segment as well as the contracted domain segment to be even (≥ 4) though it can be an odd number too. In such a case, the transformations which are described below for an even number have to be changed accordingly.

Let s_d be the starting point of the contracted domain segment $\mathcal{D}_j = \{\mathcal{D}_j(p) : p \in \mathcal{A}\}$

having b number of points. Here $p \in \mathcal{A}$ where,

$$\mathcal{A} = \{s_d,\ s_d+1, \cdots,\ s_d+\frac{b}{2}-2,\ s_d+\frac{b}{2}-1,\ s_d+\frac{b}{2},\ s_d+\frac{b}{2}+1, \cdots,\ s_d+b-2,\ s_d+b-1\}$$

The list of transformations (α_j) is presented below.

1) Identity :

$$\alpha_1(\mathcal{D}_j(p)) = \mathcal{D}_j(p); \quad \forall p \in \mathcal{A} \tag{2.7}$$

2) Second half reflection :

$$\alpha_2(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j(p); & if \quad p \leq s_d + \frac{b}{2} - 1 \\ \mathcal{D}_j(2s_d + \frac{3b}{2} - 1 - p); & if \quad p > s_d + \frac{b}{2} - 1 \end{cases} \tag{2.8}$$

3) First half reflection :

$$\alpha_3(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j(2s_d + \frac{b}{2} - 1 - p); & if \quad p \leq s_d + \frac{b}{2} - 1 \\ \mathcal{D}_j(p); & if \quad p > s_d + \frac{b}{2} - 1 \end{cases} \tag{2.9}$$

4) First half reflection and second half swapping :

$$\alpha_4(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j(2s_d + b - 1 - p); & if \quad p \leq s_d + \frac{b}{2} - 1 \\ \mathcal{D}_j(p - \frac{b}{2}); & if \quad p > s_d + \frac{b}{2} - 1 \end{cases} \tag{2.10}$$

5) First half swapping and second half reflection :

$$\alpha_5(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j(p + \frac{b}{2}); & if \quad p \leq s_d + \frac{b}{2} - 1 \\ \mathcal{D}_j(2s_d + b - 1 - p); & if \quad p > s_d + \frac{b}{2} - 1 \end{cases} \tag{2.11}$$

6) Reflection about mid point :

$$\alpha_6(\mathcal{D}_j(p)) = \mathcal{D}_j(2s_d + b - 1 - p); \quad \forall p \in \mathcal{A} \tag{2.12}$$

7) First - second and third - fourth quarter reflection :

$$\alpha_7(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j(2s_d + \frac{b}{2} - 1 - p); & if \quad p \leq s_d + \frac{b}{2} - 1 \\ \mathcal{D}_j(2s_d + \frac{3}{2}\frac{b}{2} - 1 - p); & if \quad p > s_d + \frac{b}{2} - 1 \end{cases} \tag{2.13}$$

8) Second - third quarter reflection :

$$\alpha_8(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j(s_d + \frac{b}{4} - 1 - p); & if \quad s_d + \frac{b}{4} - 1 < p < s_d + \frac{3}{4}\frac{b}{4} \\ \mathcal{D}_j(p); & otherwise \end{cases} \tag{2.14}$$

2.8.1.4 Fractal codes using GA

The main aspect of fractal based coding is to find a suitable domain segment and a transformation for a range segment. Thus the whole problem can be looked upon as a search problem. Instead of a global search mechanism GAs have been introduced to find the near optimal solution. A detailed description of GAs is already presented in Chapter 2, Section 2.3.

The number of possible domain segments to be searched is $(z-2b+1)$ (Section 2.8.1.2). The number of transformations to be searched for each domain block is 8 (Section 2.8.1.3). Thus the space to be searched consists of M elements. M is called cardinality of the search space. Here $M = 8 \times (z - 2b + 1)$. Let the space to be searched be represented by P where

$$P = \{1, 2, \cdots, (z - 2b + 1)\} \times \{1, 2, \cdots, 8\}.$$

Binary strings are introduced to represent the elements of P. The set of 2^L binary strings, each of length L, are constructed in such a way that the set exhausts the whole parametric space. The value of L depends on the values of z and b. The fitness value of a string is taken to be the RMSE between the given range segment and the estimated range segment. Note that the problem under consideration is a minimization problem. Here we are to minimize the RMSE of estimated range segment with respect to the given range segment.

Let S be the population size and T be the maximum number of iterations for the GA. Initially, S strings are selected randomly from 2^L strings, to result in an initial population for GA. The various steps of the GA, as mentioned in Chapter 2, Section 2.3 are implemented repeatedly up to T iterations. Note that the total number of strings searched up to T iterations is $S \times T$. Hence, $\frac{M}{S\,T}$ provides the search space reduction ratio for each range segment, and $(M - S\,T)$ provides the actual reduction in the search space for each range segment. Thus for n number of range segments, $(n \times (M - (S\,T)))$ would provide the total search space reduced for finding the set of transformations $\mathcal{G}$ for fractal based one dimensional signal compression.

2.8.1.5 Compression

Compression techniques practically aim at obtaining maximum data volume reduction while preserving the significant signal features upon reconstruction. Conceptually, data compression is the process of detecting and eliminating redundancies in a given data set. Redundancy may be defined as that fraction of a data which is unnecessary and hence repetitive in the sense that if it were missing the message would still be essentially complete, or at least could be completed.

Whenever the term compression is used, we need to estimate its quantitative extent. The compression ratio and redundancy are two such commonly used measures. If N_o is the size of the original file in bits and N_c is the size of the compressed signal, then the compression ratio C_R is $C_R = \frac{N_o}{N_c}$ and redundancy R_D is $R_D = (1 - \frac{1}{C_R}) \times 100\ \%$. Clearly, $R_D \rightarrow 100\ \%$ for $N_c \ll N_o$ and that indicates good compression.

We have used EEG signals of 1024 points (4 seconds epochs) each. The amplitude value of each point is a real number ranging from -10.0 to +10.0, where 10.0 denotes the brain electrical activity of about 200 microvolts. The sign +ve denotes electrical activity towards the recording electrodes, while the sign -ve denotes electrical activity away from the electrodes. For the purpose of implementation, we have converted the amplitude range to 0.0 to 20.0. Thus, 5 bits are sufficient for storing the amplitude values which are integers.

In the encoding process, for each range segment, the matched domain segment and the matched transformation have been obtained. Thus, we have to store (1) the location of the domain segment, (2) the orientation (any one of the eight transformation [Section 2.8.1.3]) of the domain segment, (3) the scaling factor and (4) the shift factor (Section 2.8.1.2). So, the number of bits stored, for a range segment, depends on the range of values of the four, above mentioned parameters as well as the size of the range segment (b). Note that the values of first two parameters are ranging from 1 to (1024-b) and 1 to 8 respectively. Discretized values for scaling and shift parameters have been considered. The process of quantizations of scale and shift parameters are same as in the case of images. Hence, 2 bits and 6 bits are required to store the scale and the shift parameters respectively. Moreover, in the case of two level partition scheme, as both parent and child range segments are considered, one more bit is required to indicate the type of range segment.

In our case, the range segment size b has been chosen to be 32 (parent) and 16 (child). The number of bits stored for both parent and child range segments becomes 10+3+2+6=21 bits. In single level scheme, $N_c = N_R \times 21$, where N_R is the number of range segments. Also, in two level scheme, $N_c = N_{PR} + (N_{PR} + N_{CR}) \times 21$, where N_{PR} and N_{CR} are number of parent range segments and child range segments respectively.

2.8.1.6 Fidelity

For validating the reliability of the compression method, the fidelity (quality) of the reconstructed signal has to be assessed. Mainly two types of performance measures, quantitative and qualitative, are used for this purpose.

The most commonly used quantitative measures are Signal-to-Noise Ratio (SNR) and the Peak-Signal-to-Noise Ratio ($PSNR$). The SNR is defined as the ratio of the root mean square (RMS) signal power to the RMS noise power, (where noise is defined as the difference between the original and reconstructed signals). Similarly, PSNR is a function of RMS signal power. The same PSNR measure has been used in the case of images. Now If $V(p)$ is the pth point of the original signal and $\hat{V}(p)$ is the same for the reconstructed signal then SNR and PSNR for one dimensional signal are defined as follows.

$$SNR = \sqrt{\frac{\sum_p V(p)^2}{\sum_p (V(p) - \hat{V}(p)^2}},$$

and

$$PSNR = -20 \log_{10} \left(\frac{\sqrt{\sum_p (V(p) - \hat{V}(p))^2}}{2^t - 1} \right),$$

where, t is the number of bits per point (pixel in the case of images) in the signal.

We have also checked the fidelity of the reconstructed signal by (a) using Cross Correlation (CC) between the actual and estimated signal and (b) examining the power spectra of both the signals.

Cross Correlation is actually the statistical correlation between two signals. We have measured the correlation between the original and estimated EEG signals. The Cross Correlation computed as

$$CC = \frac{cov(V(p), \hat{V}(p))}{\sqrt{var(V(p))\ \ var(\hat{V}(p))}},$$

where $cov(V(p), \hat{V}(p))$ is the covariance between $V(p)$ and $\hat{V}(p)$; $var(V(p))$ and $var(\hat{V}(p))$ are the variances of $V(p)$ and $\hat{V}(p)$ respectively. Cross Correlation plays an important role in judging the resemblance of two one dimensional signals. We can conclude that the reconstructed signal is very close to the original one as CC is close to unity. Note that the Cross Correlation measure has not been used in case of images. PSNR is most widely used measure in case of images.

Power spectra of the signals are used to define another quantitative measure. Apart from comparing the frequency and amplitude changes, FFT power spectra of EEG signals often convey more information [153]. This is also known as quantitative EEG or qEEG [153]. To identify EEG signal clinically, power spectrum plays an important role. Therefore FFT power spectra of both the original and the reconstructed signals have been computed for comparison. Similarities in both the power spectra mean that similar clinical conclusions can be drawn from both the signals [153]. A few statistical measurements have been taken to find the similarity between two FFT power spectra. Note that to identify EEG signals in a neural network set up, the power of FFT of EEG at each integer frequency from 1 to 30 Hz have already been considered as the input features [153]. Then, for the same purpose, input features were also reduced to three by computing the average power contained in three bands *viz.*, slow activity (1-4 Hz), medium activity (5-15 Hz) and fast activity (16-30 Hz) [113]. Thus for the present purpose of comparing two FFT power spectra, the average power in each of these three bands has been computed along with the frequency at which maximum power has occurred.The EEG power spectra have been calculated by an FFT routine from the digitized signals [153].

2.8.1.7 Implementation and Results

The proposed method has been tested on three states of EEG (wakeful, REM and slow wave sleep). In each state, the data (of four seconds duration) consists of 1024 (z) points. We have examined the performance of the proposed algorithm for two level (parent and child) partition scheme as well as single level partition scheme (parent only) using eight transformations (Section 2.8.1.3). The other parameters used are: (i) parent range segment size (b) = 32; (ii) mating pool size for each iteration of the GA (S) = 4; (iii) number of iterations for each range segment of the GA (T) = 390. The search space reduction ratio ($\frac{M}{ST}$) in each range segment is found to be about 5. The original EEG signals have already been shown in Figure 2.28. The decoded signals are shown in Figure 2.29. In Figures 2.30 and 2.31, the FFT power spectra of the original and decoded EEG signals respectively are shown.

Tests results and some statistics for a two level scheme are presented in Table 2.8. The cross-correlation, PSNR and SNR values of the original and decoded signals (Table 2.8) indicate that the performance of the coding methodology is good in all the three states. The compression ratios (Table 2.8) indicate that at least 85% reduction is achieved in all the data sets.

Table 2.8: Results of the algorithm using **two** level partition scheme

State	b	N_{PR}	N_{CR}	C_R	R_D	$PSNR$	SNR	CC
Awake	32	13	38	8.35	88.03%	51.2	0.12	0.90
SWS	32	2	60	6.91	85.52%	51.0	0.12	0.98
REMS	32	14	36	8.52	88.26%	52.9	0.10	0.95

The algorithm has also been tested using a single level scheme where the parent range segments are not subdivided in any situation. Range segments of two different sizes (viz. 16 and 32) have been selected to examine the performance of the algorithm. The values of other parameters are same as above. The results are shown in Table 2.9. The decoded signals, using single level scheme, having range segment sizes 16 and 32 are shown in Figure 2.32 and Figure 2.34 respectively. Figures 2.33 and 2.35 are showing the FFT power spectra of the decoded EEG signals in a single level scheme having

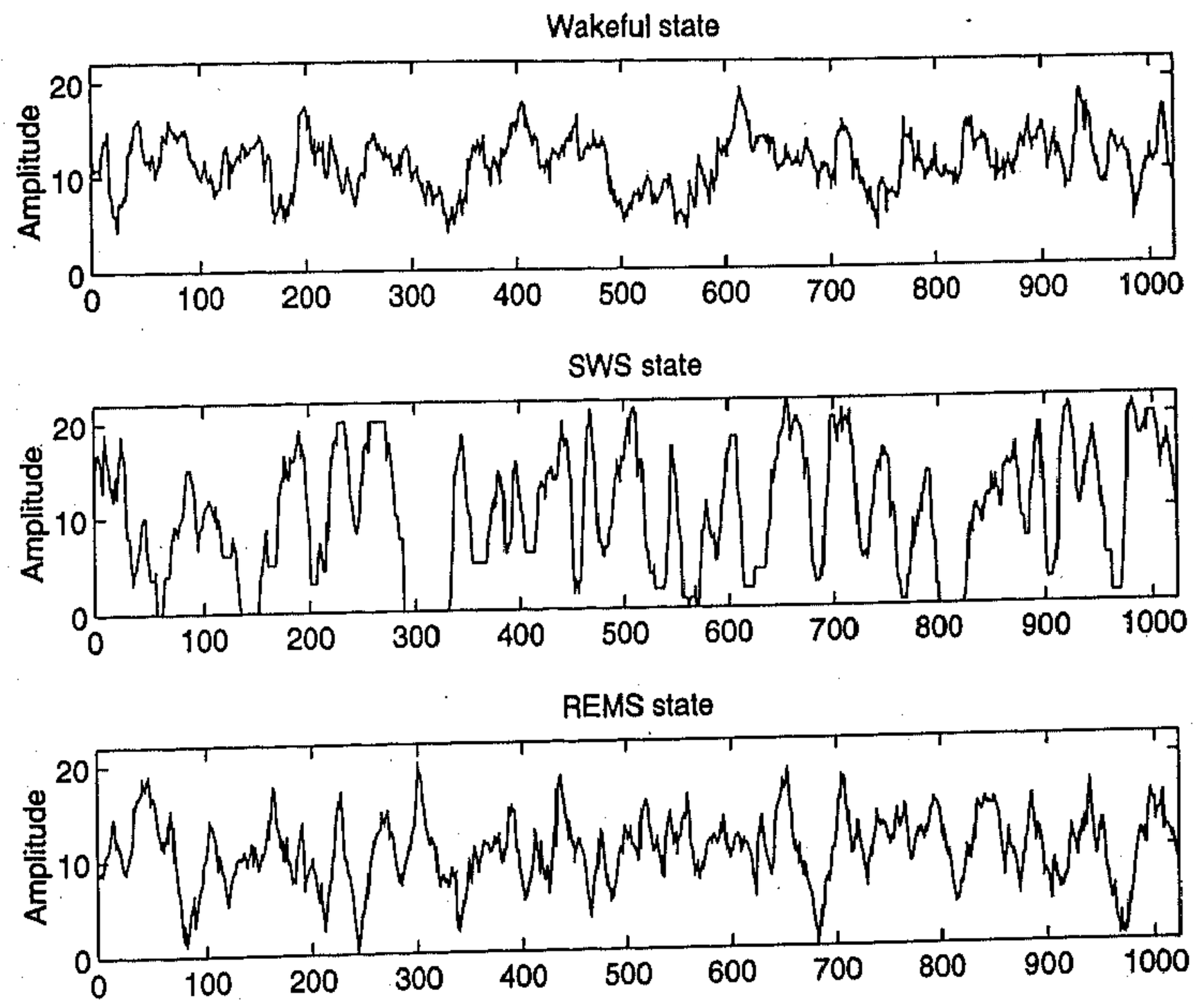


Fig. 2.29: Decoded EEG signals using two level partioning scheme in three states

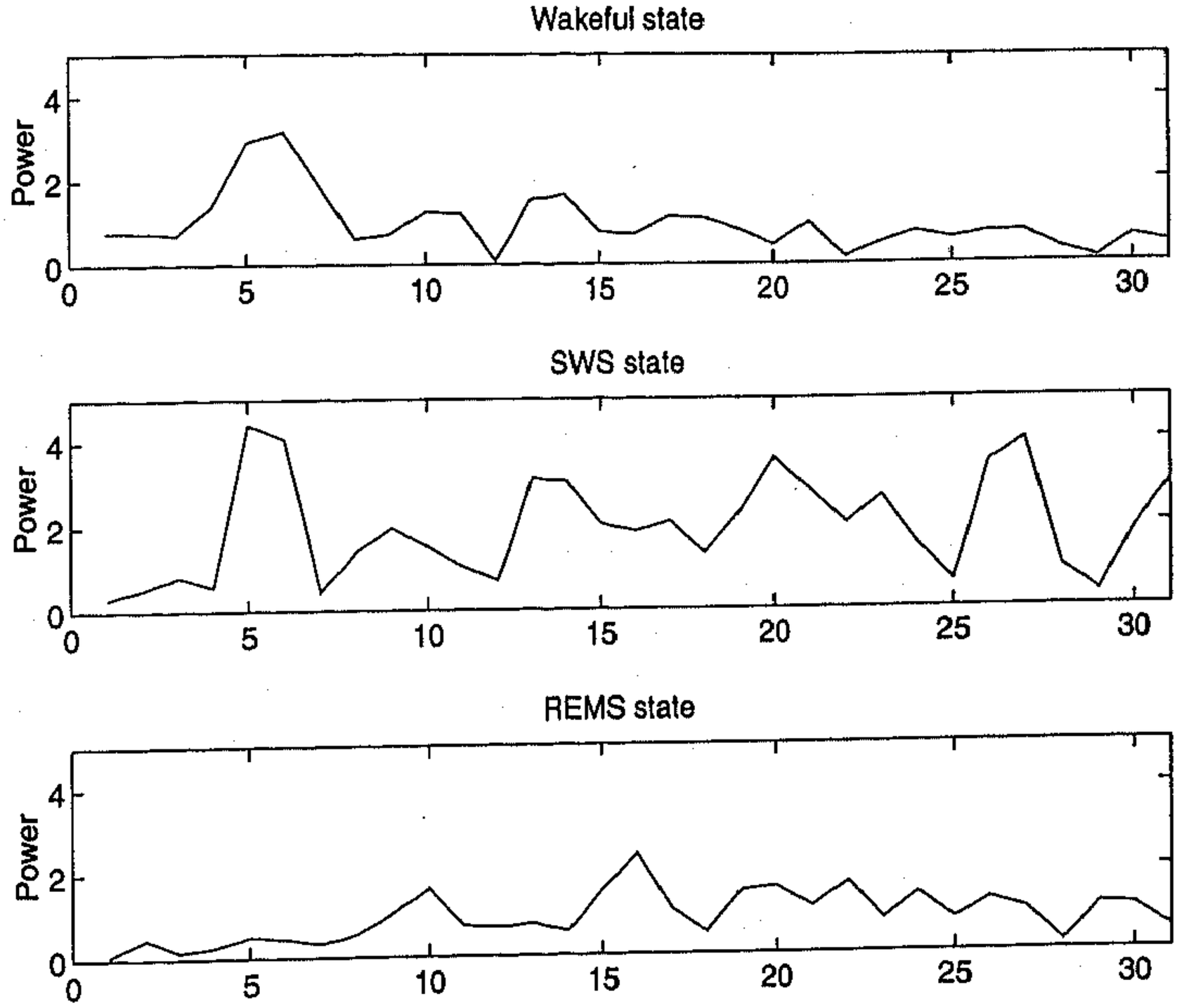


Fig. 2.30: FFT power spectra of original EEG signals in three states

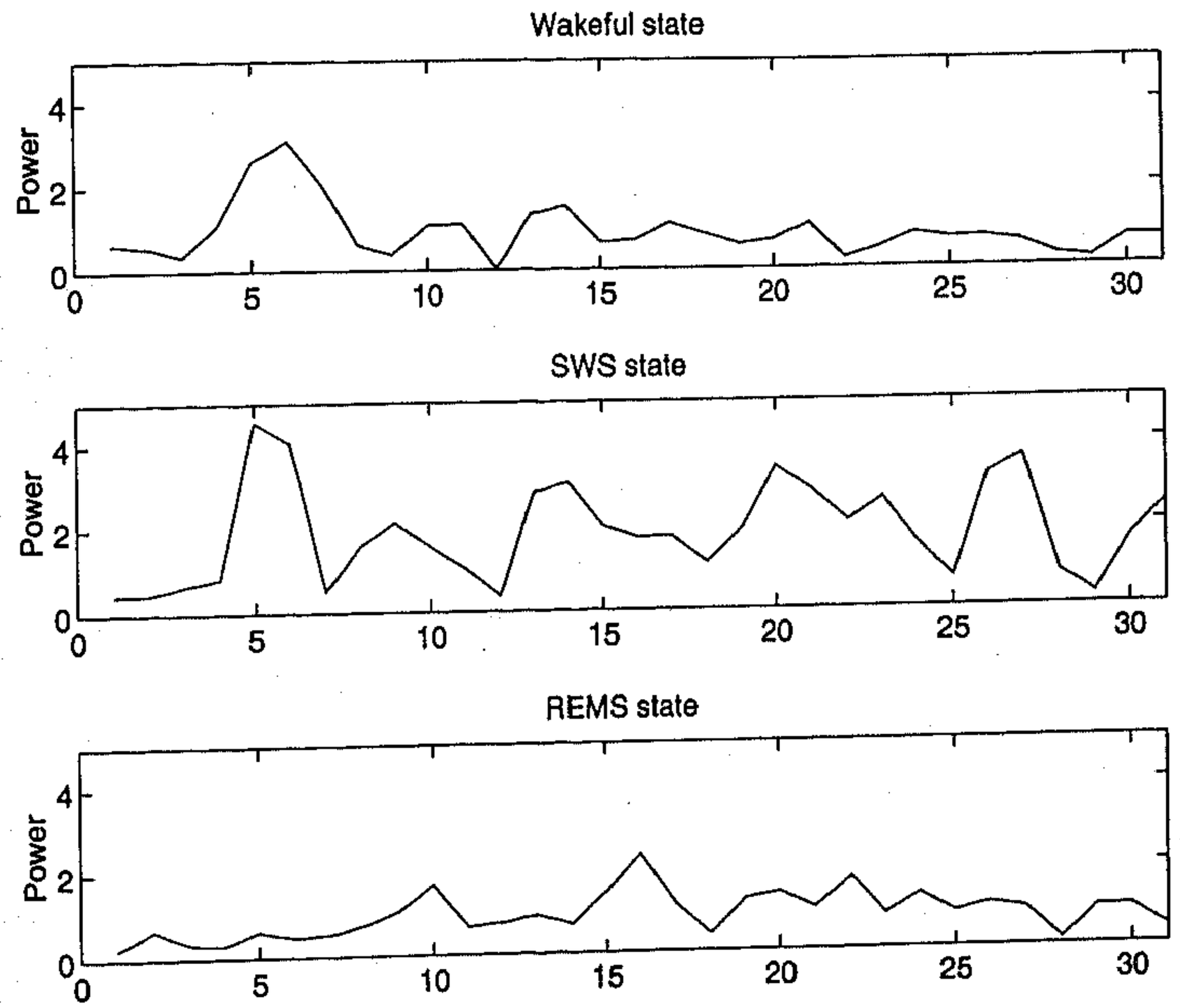


Fig. 2.31: FFT power spectra of decoded EEG signals using two level partitioning scheme in three states

range segment sizes 16 and 32 respectively.

Table 2.9: Results of the algorithm using **single** level partition scheme

State	b	N_{PR}	N_{CR}	C_R	R_D	$PSNR$	SNR	CC
Awake	16	64	Nil	6.86	85.41%	52.6	0.10	0.93
SWS	16	64	Nil	6.86	85.41%	51.0	0.12	0.98
REMS	16	64	Nil	6.86	85.41%	54.1	0.09	0.96
Awake	32	32	Nil	13.71	92.71%	49.4	0.14	0.86
SWS	32	32	Nil	13.71	92.71%	45.7	0.21	0.92
REMS	32	32	Nil	13.71	92.71%	48.2	0.18	0.85

Now for the purpose of comparing original and decoded EEG signals from FFT power spectra a few measurements have been computed as described in Section 2.8.1.6. Table 2.10 shows average power contained in each of the three frequency bands, *viz.*, 1-4 Hz, 5-15 Hz and 16-30 Hz along with the frequency at which power is maximum for EEG signals. It can be observed, from Table 2.10, that a two level partition scheme with range segment size 32 (parent) and 16 (child), and a single level partition scheme with range segment size 16 are quite satisfactory for encoding EEG signals. The decoded signals obtained using these schemes are found to be clinically same as their respective original signals. The single level scheme with range segment size 32, with a few exceptions, is also found to be clinically acceptable.

2.9 Conclusions and Discussion

An efficient fractal image compression scheme based on Genetic Algorithms has been discussed in the present chapter. The efficiency of the proposed algorithm has been established, through several experimental results on different images, by achieving reduction of the order of 20 in the search space for finding suitable affine contractive maps which approximate the given target image. At the same time, the performance of the proposed compression scheme, in terms of the compression ratio and the quality of the decoded images, is found to be comparable, many times better than that of the other existing methodologies for fractal image compression.

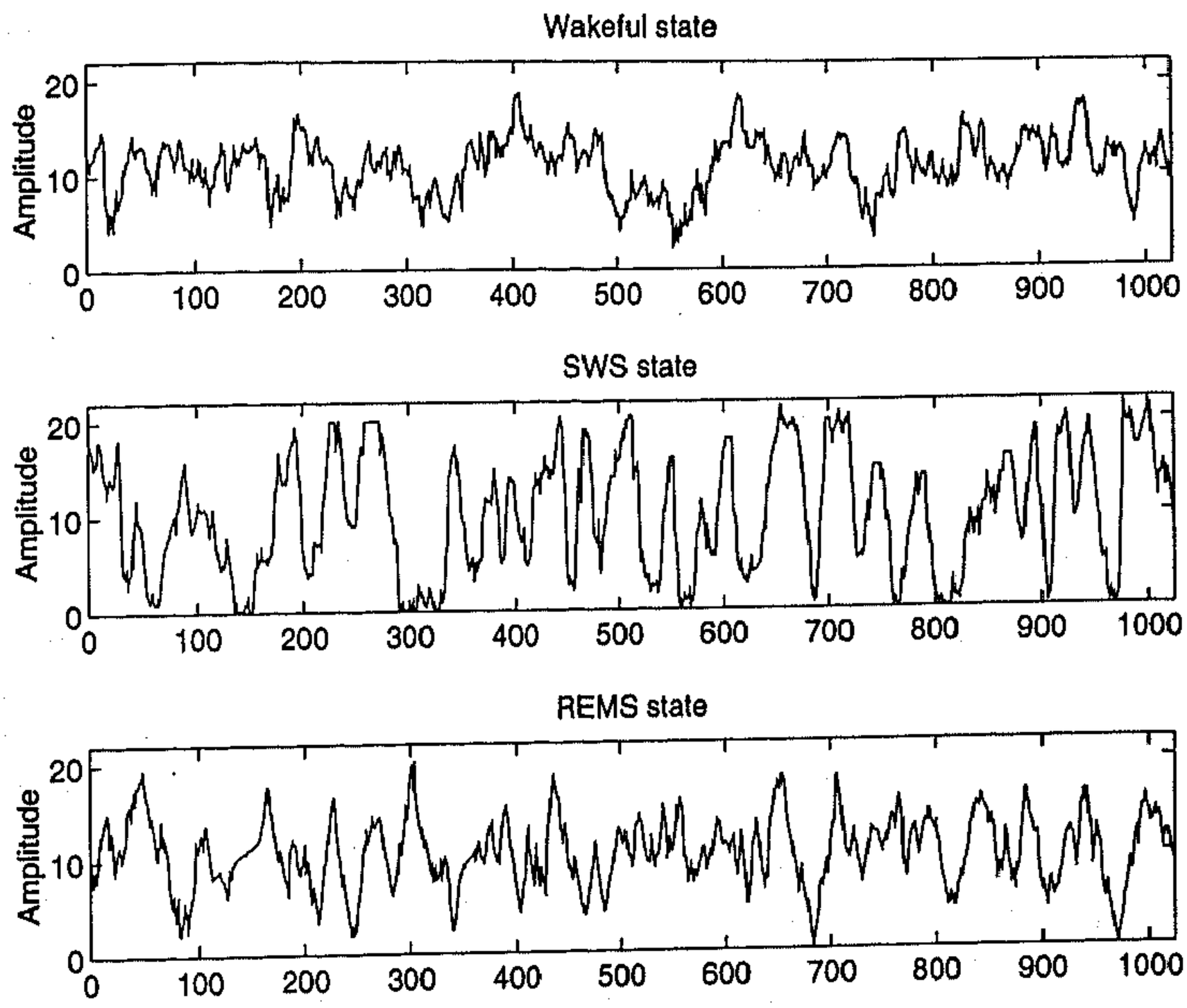


Fig. 2.32: Decoded EEG signals using single level partioning scheme with range segment size 16 in three states

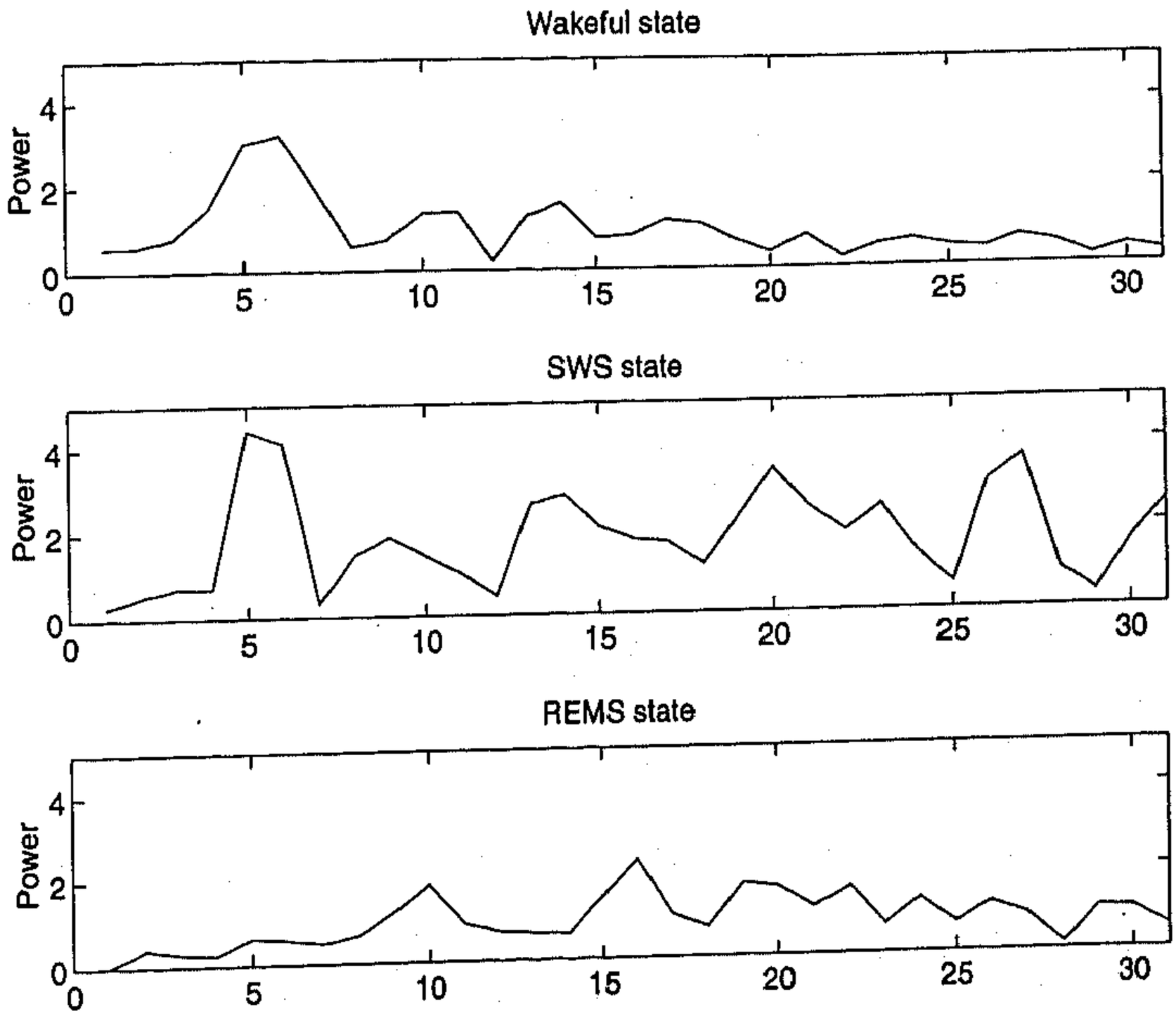


Fig. 2.33: FFT power spectra of decoded EEG signals where the decoded scheme is single level with range segment size 16 in three states

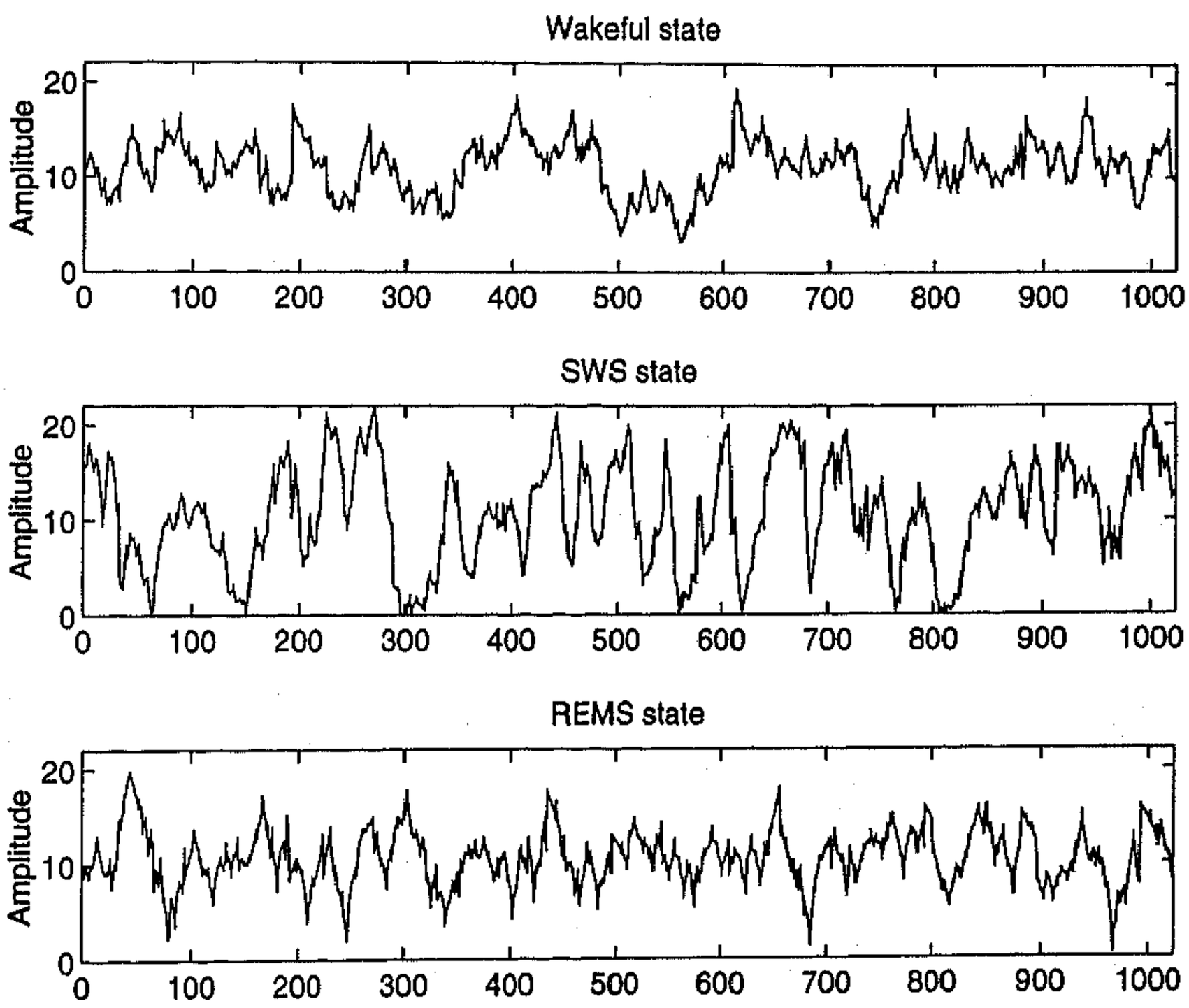


Fig. 2.34: Decoded EEG signals using single level partioning scheme with range segment size 32 in three states

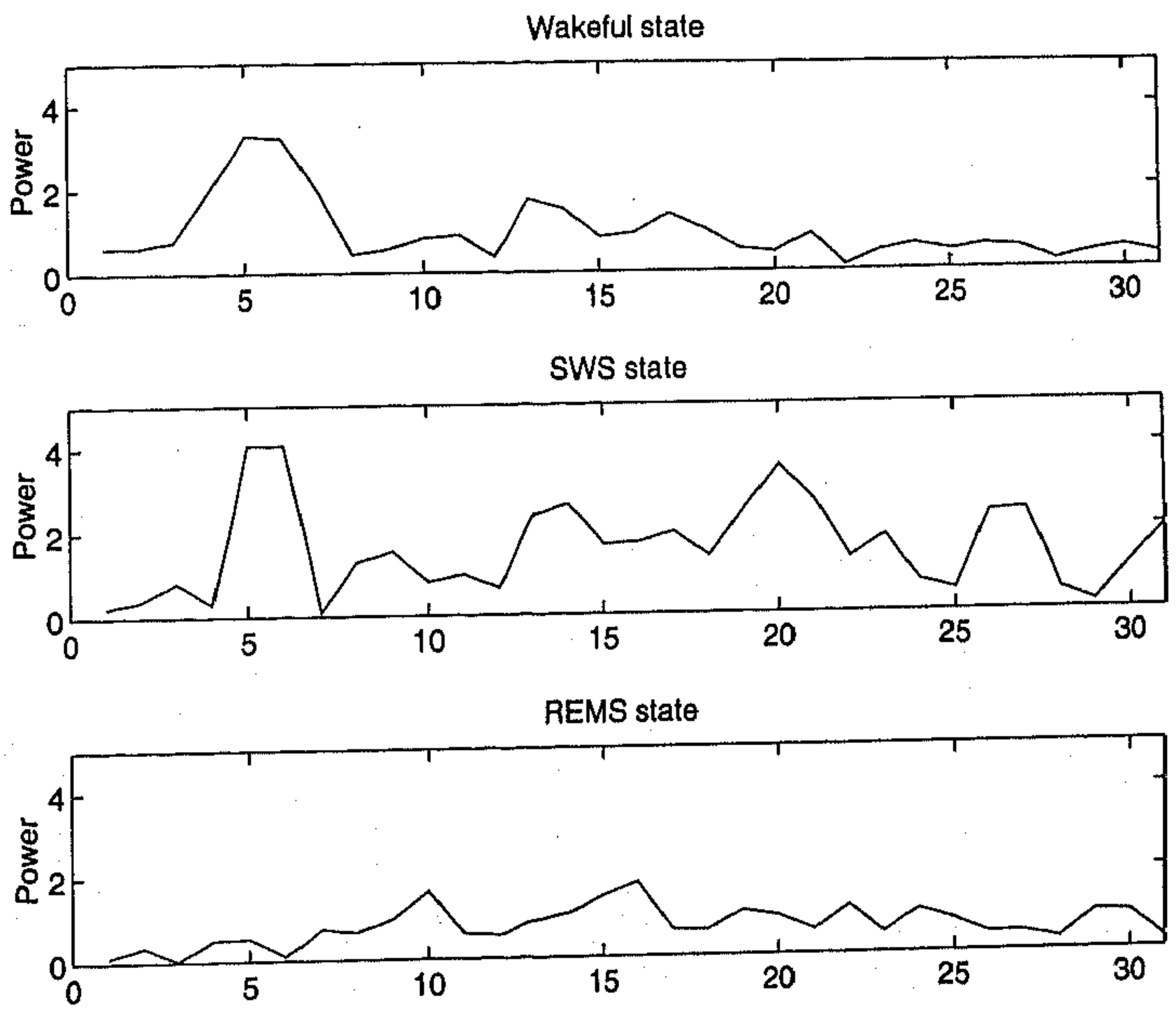


Fig. 2.35: FFT power spectra of decoded EEG signals where decoded scheme is single level with range segment size 32 in three states

Table 2.10: Some statistics obtained from the FFT power spectra

Signal	Type	Maximum		Average power within		
		Frequency	Power	1 -4 Hz	5-15 Hz	16-30 Hz
Awake	original	6	3.17	0.91	1.44	0.64
	Decoded Two level (16 & 8)	6	3.12	0.91	1.44	0.64
	Decoded single level (16)	6	3.20	0.91	1.44	0.64
	Decoded Single level (32)	5	3.27	0.91	1.44	0.64
REMS	original	16	2.38	0.24	0.79	1.15
	Decoded Two level (16 & 8)	16	2.31	0.24	0.79	1.16
	Decoded single level (16)	16	2.32	0.24	0.79	1.14
	Decoded Single level (32)	16	2.17	0.24	0.82	0.99
SWS	original	5	4.41	0.55	2.16	2.08
	Decoded Two level (16 & 8)	5	4.45	0.56	2.15	2.07
	Decoded single level (16)	5	4.40	0.55	2.14	2.08
	Decoded Single level (32)	6	4.07	0.54	1.98	1.93

The effectiveness of the GA based fractal image compression technique depends upon three factors (i) the number of points in the search space 2^L, (ii) the size of the initial population S and (iii) the number of iterations T. The number of iterations will be different for images of different sizes to achieve near optimal solutions using GAs.

PSNR is used here to measure the quality of the decoded image. There are other measures too to judge the quality of the decoded image [39]. It is easy to modify the proposed method for other measures of judging the quality of the decoded image. Root mean square error (RMSE) which is a function of mean squared error (MSE) is used as the fitness function for the set of strings. Any other suitable function, which measures the distortion between the given range block and the obtained range block, can be used as the fitness function instead of RMSE in the proposed method.

A stopping criterion of the decoding process in the present methodology can be suggested by using a threshold value on the difference between the resulting images of two successive iterations.

The threshold value for the classification of range blocks corresponds to the valley in the histogram of the variances of the pixel values of blocks. Note that it can not always be assured that the said histogram is strictly bimodal. The suitable values for the valleys in the histograms are obtained here by visual inspection for the images considered in the present chapter. One can use any one of the several thresholding methodologies (*e.g.*, [128]) for finding valleys in the histograms.

A simple technique for the classification of the range blocks is used here. However one can use other techniques too in this regard. A technique which utilizes the psychovisual properties of human visual system, has been developed recently, for the classification of the image blocks [147].

The most important factor of PIFS based schemes is the presence of similarity within the given image. In PIFS scheme, larger image blocks (domain blocks) approximates smaller image blocks (range blocks) under suitable affine transformations. Now, if a suitable affine transformation for a particular range block is not available using any domain block, then a reduction in the size of that range block would be helpful. Moreover a condensation map of the range block could be considered for the same purpose. The condensation map will make the approximation of the range block error free. A condensation map of the range block is simply the copy of the range block itself. Thus,

although the foundation of PIFS scheme on digital images is based on self similarity, this scheme performs well enough on any digital image. Hence, "self similarity" is not actually needed for applying PIFS scheme. This statement is also valid for the present GA based algorithm.

The methodology of the proposed GA based technique is not only applicable for image compression but also can be extended for general data compression. In particular, it can be adopted for coding one dimensional signals. For this purpose one needs to redefine the set of affine contractive transformations. The technique may appear to be efficient in coding very irregular curves.

The fractal based methodology for coding of one dimensional signal described in the present chapter appears to be efficient, both in terms of compression and quality, for encoding very complicated EEG signals. The decoded signals are found to be effective for clinical diagnosis as it has been observed from their FFT power spectra that same clinical conclusions can be drawn using these signals instead of their original signals.

There are only a few algorithms in the literature for encoding EEG signals. Hinrichs [70] has used an adaptive pulse code modulation scheme and achieved up to 75% of data reduction. Toraichi et al. [171] have reduced the data volume by storing the coefficients of the functions approximating the EEG waveforms. They have also divided the waveforms into different segments but of similar frequencies.

The fidelity of the reconstructed signals, in the proposed method is comparable with that of the other existing methods. On the other hand, the proposed method seems to perform better in terms of the compression ratios in comparison with [70, 171]. Here, only normal EEG during the three stages of sleep and wakefulness has been tested. This technique may be applied to the EEG in other (abnormal) states, like, epilepsy and depression. Moreover, evoked EEG responses too may be compressed using this method. It may be concluded that the fractal technique with GAs can lead to efficient and reliable compression of one dimensional signals.

Compression ratio is likely to be raised and the search space will be decreased if the number of transformations is reduced. Despite this, it has to be assured that the fidelity of the reconstructed signals should not be affected. One may use fewer number of transformations instead of eight transformations for this purpose. In such a case, the selection of transformations will play a major role.

So far, only the compression aspect of the fractal technique has been discussed in the present thesis. One question arises at this point is that whether the application of PIFS technique is restricted to image compression problem only or it can be used to perform other image processing tasks. So, investigation regarding the applicability of the fractal technique, in particular PIFS based technique, to other possible areas of image processing needs to be carried out. In this context, two algorithms, for performing image magnification and image edge extraction tasks using fractal technique are described in the following two chapters.

Chapter 3

Image Magnification

3.1 Introduction

This chapter deals with a new technique for image magnification using the theory of PIFS. The magnification task is performed using the fractal code of the image instead of the original one resulting in a reduction in memory requirement. To generate the fractal codes, Genetic Algorithm with elitist model, as described in Chapter 2, is used which greatly decreases the search for finding the suitable transformations for small image blocks.

Image magnification ideally is a process which virtually increases image resolution in order to highlight implicit information present in the original image, not evident as such. It can be looked upon as a scale transformation. Image magnification is used for various applications like matching of images captured using different sensors (having different capturing resolutions), satellite image analysis [32, 176], medical image display etc. Normally the image is represented in the form of a two dimensional array of pixel values (matrix form), and it requires large memory space. The memory requirement for storage or bandwidth requirement for transmission is greatly reduced when different coding schemes are used. The actual requirement (memory/bandwidth) is dependent on the size of the image and the method used for coding. Conventional magnification operation is generally performed on an image represented in the form of a matrix (normal form). Moreover, before applying magnification technique, any coded image has to be converted into normal form through decoding process which requires some computational cost. So it is beneficial if the magnification could be done during decoding itself. The bandwidth requirement of an image transmission system would be reduced further if an image of smaller size is transmitted but at the receiving end a magnified version is generated. The magnification technique described in this chapter is directly applicable on the coded version of the image.

So far, fractal geometry has been successfully used for image data representation [14, 38, 78] and as image processing tool [87, 139]. But, the use of either IFS or PIFS methodologies as image processing tools has not been explored much. In the present work the fractal codes, developed in Chapter 2, are used as an input to an image magnification system. Usually the codes of an image implicitly carry all the spatial information associated with the image. Besides the spatial information, the fractal codes carry the information of the similarities present in the image. This similarity property

is also exploited in the proposed image magnification technique. The similarity here means that larger image blocks can approximate smaller image blocks under suitable transformations. The said magnification scheme is nothing but a modified decoding scheme of fractal codes which gives rise to a magnified version of the original image.

The simplest solution of fractal image magnification is to decode the code to any resolution. The obvious artifacts, the magnified image gets out of this, is largely due to blocking effects. The present method is an acceptable solution to avoid the blocking effects in the magnified image. The proposed algorithm moreover is using fractal code of an image as the input instead of the original image. There is no need to store the original image anymore or to decode the image from its code. This will lead to reduce the cost of memory requirement for performing magnification task. Thus, reduction in the memory requirement and obtaining a quality magnified image by reducing blocking artifacts are two significant advantages of the proposed algorithm.

Some of the popular techniques of digital magnification of images are nearest neighbor, bilinear and bicubic interpolations [32, 88, 136, 159, 176]. All these techniques are based upon surface interpolation. An interpolation technique finds a functional relationship between two sets of values such that an unknown value of one set can be determined from the points belonging to the other set. A description of interpolation and approximation techniques in the context of image magnification has already been provided in Appendix A. Image magnification techniques using interpolation, usually, exploit the local information ignoring the global information. In these cases, some distortions due to blocking, blurring and ringing are observed in magnified images after a certain extent of expansion. In the proposed scheme, the magnification task has been performed by using fractal codes where both the local and the global information are used. The scheme results in sharper expanded images where the distortions due to ringing and blurring are found to be less. The performance of the present algorithm is compared with the nearest neighbour technique and linear interpolation technique. The performance of the suggested method is found to be better compared to both the cases. A comparison of the present magnification method with the method of zooming using fractals suggested by Polidori and Dugelay [141] has also been considered.

For any magnification technique, the distortion due to blocking, which is a local phenomenon, is very usual. A widely used distortion measure for quantifying the distortion between two images having equal size, is the mean squared error (MSE) or some other

form of it. MSE is a global measure which fails to represent properly the local distortion due to blocking. But the blocking effects are very much sensitive to the human visual system. So, to quantify the distortions in the edges, a new distortion measure is introduced. This measure is defined on the edge images having unequal sizes and is found to be suitable to serve the present purpose of judging the performance of a magnification algorithm.

In the process of magnification, the magnified image should be visually similar to the original one. In this context, we have also proposed a similarity criterion based on just noticeable difference (JND). As the sizes of the magnified image and the original image are different, the similarity between them can't be judged by inspecting the pixel values alone. Hence the JND based scheme is proposed.

Theory and key features of image magnification using PIFS methodology are outlined in Section 3.2. The methodology of using fractal codes for image magnification of a given image is described in Section 3.3. A new fidelity criterion to judge the performance of the proposed algorithm is discussed in Section 3.4. JND based similarity criterion is discussed in Section 3.5. Section 3.6 presents implementation and the results. Discussion and some conclusions are provided in Section 3.8.

3.2 Image Magnification Using PIFS

Let I be a given image having size $z \times z$ and the range of gray level values be $\{0, 1, 2, \cdots, l-1\}$. The image is partitioned into n non overlapping squares of size, say $b \times b$, and let this partition be represented by $\mathcal{N} = \{\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_n\}$. Each $\mathcal{R}_i$ is named as range block where, $n = \frac{z}{b} \times \frac{z}{b}$. So, we have,

$$I = \bigcup_{i=1}^{n} \mathcal{R}_i.$$

Now let $\mathcal{M}$ be the collection of all possible blocks (within the image support) which are of size $2b \times 2b$ and let $\mathcal{M} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m\}$. Each $\mathcal{D}_j$ is named as domain block with $m = (z - 2b + 1) \times (z - 2b + 1)$.

Let the set of transformations $\mathcal{F}_j$ be such that

$$\mathcal{F}_j = \{f \ : \ \mathcal{D}_j \to \mathbb{R}^3; \ \ f \text{ is an affine contractive map}\}$$

Now, for a given range block $\mathcal{R}_i$, the map $f_{i|j} \in \mathcal{F}_j$ be such that

$$\rho(\mathcal{R}_i,\ f_{i|j}(\mathcal{D}_j)) \ \leq\ \rho(\mathcal{R}_i,\ f(\mathcal{D}_j))\ \ \forall f \in \mathcal{F}_j,\ \forall\ j,$$

where ρ is a suitably chosen distance measure.

Now let k be such that

$$\rho(\mathcal{R}_i,\ f_{i|k}(\mathcal{D}_k)) \ =\ \min_j\{\ \rho(\mathcal{R}_i,\ f_{i|j}(\mathcal{D}_j))\ \} \tag{3.1}$$

Also, let $\quad f_{i|k}(\mathcal{D}_k) = \widehat{\mathcal{R}}_{i|k}.$

Here $\mathcal{K}_i\ =\ \{\mathcal{D}_k, f_{i|k}\}$ is called fractal code for $\mathcal{R}_i$ and the set $\mathcal{K}\ =\ \{\mathcal{K}_i\ ,\ i = 1, 2, \cdots, n\}$ is called the PIFS of the given image I.

Thus we have,

$$\rho(\ \bigcup_{i=1}^{n}\mathcal{R}_i,\ \bigcup_{i=1}^{n}\widehat{\mathcal{R}}_{i|k}\ )\ =\ \rho(\ \bigcup_{i=1}^{n}\mathcal{R}_i,\ \bigcup_{i=1}^{n}f_{i|k}(\mathcal{D}_k\ )\ )\ \ \leq\ \ \epsilon_1, \tag{3.2}$$

where ϵ_1 is a small positive quantity.

The affine contractive transformation $f_{i|k}$ is constructed using the fact that the gray values of the range block are scaled and shifted versions of the gray values of a domain block. The transformation $f_{i|j}$ is such that $f_{i|j}(\mathcal{D}_j)$ approximates $\mathcal{R}_i$. $f_{i|j}$ consists of two parts, one for spatial information and the other for information of gray values. The first part indicates which pixel of the range block corresponds to which pixel of domain block. The second part is to find the scaling and shift parameters for the set of pixel values of the domain blocks to the range blocks.

The first part denotes the shuffling the pixel positions of the domain block and can be achieved by using any one of the eight transformations on the domain blocks [80, 121]. Once the first part is fixed, second part is estimation of a set of values (gray values) of range blocks from the set of values of the transformed domain blocks. These estimates can be obtained by using the least square analysis of two sets of values [118, 121].

The second part is obtained using least square analysis of two sets of gray values once the first part is fixed. Moreover the size of the domain blocks is double that of the range blocks. But, the least square (straight line fitting) needs point to point correspondence. To overcome this, one has to construct contracted domain blocks such that the number of pixels in the contracted domain blocks become equal to that of range blocks. The

contracted domain blocks are obtained by adopting any one of the two techniques. In the first technique, the average values of four neighboring pixel values in a domain block are considered as the pixel values of the contracted domain blocks [80]. In the other scheme, a contracted domain block is constructed by taking pixels from every alternative row and column of the domain block [121]. A detail description of these schemes will be presented in Chapter 4. In the present case we have adopted the later one.

Thus $f_{i|k}$ can be looked upon as mixture of two transformations, $f_{i|k} = t_{i|k} \circ \mathcal{C}$, where, $\mathcal{C}$ is contraction operation and $t_{i|k}$ is transformation for rows, columns and gray values.

Now to perform magnification task using the given image I, one needs to define a magnification operator, say, $\mathcal{O}$ such that

$$\rho(\bigcup_{i=1}^{n}\mathcal{R}_i, \bigcup_{i=1}^{n}\mathcal{O}(\mathcal{R}_i)) \leq \epsilon_2 \tag{3.3}$$

where ϵ_2 is a small positive quantity.

Here $\mathcal{O}$ is a magnification operator which maps an image to another image of size greater than that of input image.

From (3.1) we have,

$$\rho(\mathcal{R}_i, \widehat{\mathcal{R}}_{i|k}) \leq \epsilon_{i1} \tag{3.4}$$

Where ϵ_{i1} is a small positive quantity. Now by (3.2), (3.3) and (3.4 we have,

$$\rho(\bigcup_{i=1}^{n}\mathcal{R}_i, \bigcup_{i=1}^{n}\mathcal{O}(\widehat{\mathcal{R}}_{i|k})) \leq \epsilon_3 \tag{3.5}$$

where ϵ_3 is a small positive quantity. Again, we have,

$$\widehat{\mathcal{R}}_{i|k} = f_{i|k}(\mathcal{D}_k) = (t_{i|k} \circ \mathcal{C})(\mathcal{D}_k).$$

So,

$$\rho(\bigcup_{i=1}^{n}\mathcal{R}_i, \bigcup_{i=1}^{n}(\mathcal{O} \circ t_{i|k} \circ \mathcal{C})(\mathcal{D}_k)) \leq \epsilon_3. \tag{3.6}$$

Now, reconstruction of images using the magnification operator $\mathcal{O}$ should be an inverse of the contraction operation using the operator $\mathcal{C}$. So, by (3.5) and (3.6)

$$\rho(\mathcal{O}(\widehat{\mathcal{R}}_{i|k}), t_{i|k}(\mathcal{D}_k)) \leq \epsilon_4. \tag{3.7}$$

Hence, by (3.5),

$$\rho\left(\bigcup_{i=1}^{n}\mathcal{R}_i\ ,\ \bigcup_{i=1}^{n}t_{i|k}(\mathcal{D}_k)\right)\leq\epsilon_5. \tag{3.8}$$

Both ϵ_4 and ϵ_5 are small positive quantities.

Remarks :

1. From equation (3.8), it is clear that there is no need of constructing the magnification operator $\mathcal{O}$, only the second part of the fractal codes has to be applied on the domain block to get an image which is very close to the given image I and this image has size $2z \times 2z$.

2. As the process of magnification ends up with a magnified image, the distance measure (ρ), used in this set up, should be selected in such a way that it can compute the distortion between two images with unequal sizes. Note that, though MSE can be chosen as a suitable distance measure for image compression, the same measure is not suitable for the present magnification task. MSE can measure the distortion between two images having equal sizes. Thus a proper selection of ρ is needed for the present case.

3. The above results are also valid in case of condensation maps included in the PIFS codes. It has already been discussed in Chapter 2 that condensation maps, considered here, are contractive maps. Hence, magnification using condensation maps, when ever it is required, can be achieved.

In the next section, the methodology for obtaining magnified image of a given image is presented.

3.3 Methodology for Fractal Based Image Magnification

The most important factor involved in a magnification task is the order of magnification. In the present case the order of magnification is defined as follows. Let an image I having size $z \times z$ is used as an input to a magnification system. Now if the output

image $\hat{I}$ (say) is of size $kz \times kz$, then k is called the order of magnification. Usually k is taken to be an integer, though k can be a fraction (> 1) too. In this thesis the term magnification factor is also used to indicate the order of magnification. So far we have discussed how to apply the fractal codes to get a magnified image which is magnified by a factor 2. On successive applications of this proposed algorithm, magnification by factor 4, 8, 16 etc. can also be achieved. But the first task is to obtain the fractal codes for a given image.

3.3.1 Construction of PIFS Codes for Magnification

The size of a range block plays an important role in image compression as well as magnification. If small blocks are taken, the finer details of the image are preserved and restored in the decompressed image but the compression ratio will be less. On the other hand more compression will be achieved, at the cost of finer image details, if larger range blocks are considered. Thus a trade off has to be made to get good quality decompressed image as well as considerable amount of compression. But the main task in magnification is only to restore all the image details and almost no emphasis is given on the amount of compression achieved. So, in this case, small range blocks are considered to keep track with every minute details of the original image.

In the proposed algorithm, to obtain the fractal codes of small range blocks of a given image, the blocks are first classified into two groups using a simple classification scheme [121] [see Chapter 2]. The groups are formed according to the variability of the pixel values in the blocks. If the variability of a block is low *i.e.*, if the variance of the pixel values in the block is below a fixed value, called threshold, we call the block as smooth type range block. Otherwise we call it a rough type range block. The threshold value to separate the range blocks into two types is obtained from the valley in the histogram of the variances of pixel values of the blocks. Every pixel value in a smooth type range block is replaced by the mean of all pixel values. So, it is enough to store only the discretized mean value. On the other hand for each rough type range block, the appropriately matched domain block as well as appropriately matched transformation from eight possible isometric transformations [80] have to be searched out. To solve this search problem the GA based technique as described in Chapter 2 is adopted.

In the next subsection the technique for successive magnifications has been described.

3.3.2 Successive Magnification

Fractal block coding, used here, is a lossy compression. One expects a small error would be introduced in case of magnification by order 2. The recoding of this approximation will introduce further error, resulting high degree of divergence of the magnified image from its the original. Moreover, the recoding of the approximation at each step is a time consuming process as fractal coding itself is time consuming. Thus, to reduce the error as well as the computational time at each step, the successive magnification methodology is introduced.

Here the term successive magnifications implies dividing the task of magnification by a factor, which is a power of 2, into several steps. In particular magnification of an image by a factor 2^n (n is a positive integer) is divided into n steps. In the first step, magnified image is obtained from PIFS code considering $n = 1$. Then in the next step, *i.e.*, in the case of $n = 2$, magnification task is performed using the PIFS code and the information regarding the already obtained magnified image in the previous step. Incrementing the value of n at each step and proceeding in a similar way, the required magnified image is obtained.

In the case of successive applications of the algorithm, the PIFS codes need not be computed afresh. The location of the matched domain block is kept fixed to speed up the method of magnification. It is true that one may expect a little degradation in the quality of the magnified image. But fresh encoding at each step will require a huge computational time to find fractal code even after introduction of sophisticated search mechanism such as GA (Chapter 2). Successive magnification has been proposed to make a trade-off in between computational time and quality of the magnified image.

The code used in each step is obtained by modifying the code of the previous step. In particular, the transformations $t_{i|k}$ are modified by using the image information that has already been obtained in the previous step. The locations of the appropriately matched domain blocks remain same in all the steps. Only the sizes of the domain blocks are increased in the modification process. Thus, in particular, only the gray level transformation in $t_{i|k}$ is to be modified. The gray level transformations are obtained using least square technique. In this technique a straight line is fitted with two sets of gray level values of which one is from range block and other is from contracted domain block. In the successive magnification schemes these two sets are enlarged. These

enlarged sets are divided into several parts and separate straight lines are fitted for each part using the same least square technique. The sets are divided into 4 parts for achieving magnification by a factor 4 and divided into 16 parts for the magnification by a factor 8, and so on. So, the number of fractal codes, in a step, becomes 4 times larger than its counter part in the previous step. Thus, it is enough to find the fractal codes for a factor 2 and they need to be modified accordingly for achieving magnification greater than two.

Though in the present chapter we have discussed the methodology of magnification by orders which are powers of 2, the method can be extended for magnification by any order. A discussion on the possible extensions of the algorithm has been presented in concluding section (Section 3.8).

In the next two sections we shall discuss the evaluation criteria to judge the performance of the proposed algorithm.

3.4 Fidelity Criterion

It is necessary to judge the performance of the proposed fractal based image magnification algorithm. For this purpose one has to measure the distortion between the given image and the reconstructed image. To quantify the amount of distortion, the widely used distortion measure is peak signal to noise ratio (PSNR) which is a function of Mean Squared Error (MSE). MSE or PSNR examines the similarities between two images. But MSE is a size dependent measure *i.e.*, the two images, under consideration, should have the same size. Moreover it is a global measure and it finds the average pixel to pixel difference. It does not accurately indicate the large and significant local distortions due to blocking or blurring [39] as it deals with the average distortions. But the blocking effects are very much sensitive to the human visual system. So, one has to think of a size independent measure which reflects local as well as global distortions and judges the performance of magnification methodology. A new fidelity criterion whose performance is found to be suitable for the present case is introduced. This measure finds the distortion between the edge images of the given image and the magnified reconstructed image. The overall performance of the proposed algorithm is found using this new distortion measure.

3.4.1 Edge Based Distortion Measure

The images that are obtained from the codes usually have specific artifacts such as blocking, ringing and blurring. Actually these artifacts are reflected more prominently in the high frequency component of the image and are very sensible to the human visual system [86]. So, in our proposed distortion measure, we have tried to measure the dissimilarities in edge pattern. It is expected that the edge pattern of the original image will be preserved, in a magnified way, in the magnified image. In other words, the distribution of the edge pixels of a magnified image should be almost similar to that of the original image. The proposed error criterion is based on the distributions of edge pixels in two images. For simplicity, only the vertical and horizontal edges have been considered here. Note that, the proposed distortion measure though is suitable for the present case, may not be useful for measuring the distortion between two images in a different context.

Both the images are first partitioned into blocks proportional to their respective sizes in such a way that both images contain equal number of blocks. The error is then measured block wise and finally the average error is computed. To detect the edges of each block we have used the scheme suggested by Ramamurthi et al. [146]. The edge blocks consist of value "0" and "1" where, "1" represents the presence of edge. Now it is expected that the original and the magnified blocks should have the same type of edge distributions. In other words the expected run of 1's present in both the blocks should be same if normalized by their respective sizes. Thus the distortion measure is defined by the difference between the normalized "expected run" of 1's present in the given image and in the magnified reconstructed image. The vertical and horizontal edges are considered separately and then averaged to give rise to the final distortion measure of a block. The algorithm for calculating the value of the proposed distortion criterion is described below.

3.4.1.1 Description of the Algorithm

Step 1 : Partition the images, I_1 and I_2 (with size of I_1 less than size of I_2) into square blocks such that the number of partitions is same in both the images. Let $p_1 \times p_1$ and $p_2 \times p_2$ (with $p_1 < p_2$) be the sizes of the square blocks for the images I_1 and I_2 respectively. Let these blocks be $B_{11}, B_{12}, \cdots, B_{1n}$ and $B_{21}, B_{22}, \cdots, B_{2n}$.

Step 2 : From $B_{ij}, i = 1, 2$ and $j = 1, 2, \cdots, n$ compute gradient matrices or the edge images. Let G^h_{ij} and G^v_{ij} be respectively the horizontal and the vertical gradient matrix. The elements of the gradient matrices are all either 0 or 1. The gradient matrices are defined as follows

$$\begin{aligned} G^h_{ij}(m,n) &= 0 \quad if \quad \frac{|g_{m,n} - g_{m,n+1}|}{\frac{g_{m,n}+g_{m,n+1}}{2}} < T \\ &= 1 \quad if \quad Otherwise. \end{aligned}$$

and

$$\begin{aligned} G^v_{ij}(m,n) &= 0 \quad if \quad \frac{|g_{m,n} - g_{m+1,n}|}{\frac{g_{m,n}+g_{m+1,n}}{2}} < T \\ &= 1 \quad if \quad Otherwise. \end{aligned}$$

Here $g_{m,n}$ = Gray level value of $(m,n)th$ pixel in a block and T = A prefixed threshold value.

Step 3 : Find the expected run of 1's present in both horizontal and vertical directions in both G^h_{ij} and G^v_{ij}. Let L be the random variable denoting the number of run of 1's in a particular gradient matrix in a particular direction. Compute $E^{hh}_{ij}(L)$, $E^{hv}_{ij}(L)$, $E^{vh}_{ij}(L)$, $E^{vv}_{ij}(L)$. Here expected run of 1's is defined as

$$E(L) = \sum_k k \frac{Number\ of\ times\ the\ run\ of\ length\ k\ appears}{Total\ number\ of\ runs\ (of\ all\ possible\ lengths)\ present}.$$

Now compute

$$E_{ij}(L) = \frac{E^{hh}_{ij}(L) + E^{hv}_{ij}(L) + E^{vh}_{ij}(L) + E^{vv}_{ij}(L)}{4}.$$

Step 4 : Normalize $E_{ij}(L)$ by respective block size.

$$\begin{aligned} E_{ij}(L) &= \frac{E_{ij}(L)}{p_1 \times p_1} \quad if \quad i = 1 \\ &= \frac{E_{ij}(L)}{p_2 \times p_2} \quad if \quad i = 2 \end{aligned}$$

Step 5 : Compute the final error measure E between the two given images I_1 and I_2. E is defined as

$$E = \frac{1}{n} \sum_{j=1}^{n} \{E_{1j}(L) - E_{2j}(L)\}^2.$$

The next section discuses the JND based similarity criterion.

3.5 JND Based Similarity Criterion

Just noticeable difference (JND) measures the amount of change in gray value of a pixel in comparison of its surrounding pixels. Usually, JND is used to evaluate the edges present in an image [93]. Here we have proposed a similarity measure based on JND to judge the similarity between two images having unequal sizes. In particular, the proposed similarity measure judges the performance of the proposed PIFS based image magnification technique. JND is basically the difference in contrast of an object from its background and it plays an important role in human visual system. The human visual system (HVS) model [39] deals mainly with three factors, the luminance level, spatial frequency and signal content. Out of these, the perceived luminance is a nonlinear function of incident luminance. According to Weber's Law [24], if the luminance $(L_B + \triangle L)$ of an object is just noticeably different from its background luminance L_B, then $\triangle L/L_B$ = constant. Therefore, the just noticeable difference (JND) $\triangle L$ increases with the increase in L_B.

In the present case, we have developed a criterion based on JND, to judge the similarity between two images, which are unequal in their size. In particular, $\triangle L$ is calculated for every pixel in each of the two images under consideration. The average of $\triangle L$ values is then calculated for each of the images. Percentage of similarity present is computed using these average values of $\triangle L$. We call this percentage of similarity based on JND as JND similarity. Note that, "JND similarity is close to 100" implies that "complete similarity is present between the two images". The computation of L_B and $\triangle L$ are carried out as described in [93].

3.6 Implementation and Results

To find the PIFS code for a given image, the search is to be made for all possible domain blocks as well as eight possible isometric transformations [80]. Genetic Algorithm is used to reduce the search space and time. The search space reduction is achieved since near optimal solutions are usually satisfactory and, intuitively, solutions which are far away from the expected are rejected in a probabilistic manner. This is the reason for GA to perform well for optimization problems. The good performance of GAs to find fractal codes of a given image has already been shown in Chapter 2. The results are

quite satisfactory and at the order of 20 times reduction in the search space is achieved.

For the specific implementation of the proposed algorithm, a portion of the original "Lena" image (Figure 3.1) is treated as the original input image. The input image is a 128×128, 8 bit/pixel image. The GA based technique of Chapter 2 is applied to generate the fractal codes. Moreover, a simple classification scheme (see Chapter 2) for range blocks have been adopted to retain the regions where the gray level variation is minimum. In the classification scheme, the range blocks are grouped into two classes *viz.*, "smooth" and "rough". Every pixel value of a smooth range block is replaced by the average of all the pixel values. For each rough type range block, the GA based technique of Chapter 2 is used to find fractal codes. In the case of magnification algorithm, small range blocks of size 2×2 are considered for the computation of fractal codes. It is true that compression ratio will be reduced by considering smaller range blocks but the finer details of rough type range blocks will be retained. The main aim of a magnification task is to magnify the image keeping all the image details. So, we have considered range blocks of size 2×2. Now, using the obtained fractal codes in the way described in Section 3.3, an image of size 256×256 is reconstructed. The reconstructed image is two times magnified than the original image. This image is found to be very close to the original image which is judged by the error measure and the similarity measure as described in Section 3.4.1 and 3.5 respectively. The fractal codes are then modified stepwise, as described in Section 3.3.2, to get the images which are 4 times and 8 times magnified than the original one. In each step, the error, in comparison to its previous step, is measured successively. Also the similarities of magnified images are judged, successively, by the JND based similarity criterion.

The proposed algorithm is compared with the nearest neighbour technique for image magnification in terms of proposed distortion measure and similarity measure. Nearest neighbour is probably one of the simplest methods of digital magnification available. Given an image of size $z \times z$, to magnify it by a factor k, every pixel in the new image is assigned the gray value of the pixel in the original image which is nearest to it. This is equivalent to repeating the gray values $k \times k$ times to obtain the magnified image. The resultant image for large magnification factors will have prominent block like structures due to lack of smoothness. The present method is also compared with linear interpolation technique of image magnification [65]. A pixel of a magnified image is estimated from two consecutive pixels of the given image using interpolation. Usually, blocking

artifacts are reduced while contrast smoothing is visible in the magnified images obtained using linear interpolation.The other techniques of digital image magnification using interpolation are based cubic or bicubic interpolations [65, 88, 136, 159].

The proposed algorithm has also been implemented on a part of the "Low Flying Aircraft" (LFA) image, "Seagull" image and "Girl" image. All the images are of size 128×128 and range of gray level values 0 to 255. Other parameters of the algorithm are kept fixed as in the case of "Lena" image. All the results obtained are presented in Table 3.1 and Table 3.2.

Table 3.1: The results obtained in terms of **distortion** of the Image magnification Algorithms

Input image	Reconstructed Image Statistics										
	MF	Distortion			MF	Distortion			MF	Distortion		
		Fractal	LI	NN		Fractal	LI	NN		Fractal	LI	NN
Lena	2	1.18	1.52	1.62	4	1.37	2.20	3.11	8	1.24	1.25	6.15
LFA	2	2.32	2.79	2.37	4	2.85	4.20	4.59	8	2.43	2.95	9.11
Seagull	2	2.16	2.35	2.28	4	2.57	3.64	4.41	8	2.48	2.63	8.76
Girl	2	2.01	2.65	2.42	4	2.51	4.37	4.69	8	2.49	2.85	9.31

MF=Magnification Factor, LI=Linear Interpolation and NN=Nearest Neighbour

Table 3.2: The results obtained in terms of **similarity** of the Image magnification Algorithms

Input image	Reconstructed Image Statistics										
	MF	Similarity (%)			MF	Similarity (%)			MF	Similarity (%)		
		Fractal	LI	NN		Fractal	LI	NN		Fractal	LI	NN
Lena	2	63.47	57.41	64.39	4	57.75	54.87	49.06	8	54.50	54.61	44.38
LFA	2	82.05	58.94	68.22	4	72.46	54.20	51.83	8	64.40	52.86	47.03
Seagull	2	78.12	68.98	79.22	4	61.49	58.70	58.19	8	59.20	55.47	49.14
Girl	2	65.77	59.63	65.83	4	58.12	54.07	49.20	8	55.23	53.46	43.89

MF=Magnification Factor, LI=Linear Interpolation and NN=Nearest Neighbour

The original and decoded images of "Lena" are shown in Figures 3.1 and 3.2 respectively. Figure 3.3, 3.4 and 3.5 are respectively the two times, four times and eight times magnified images of "Lena" using the proposed fractal based technique while Figures 3.7, 3.9 and 3.11 are respectively two times, four times and eight times magnified images of "Lena" using nearest neighbour technique. Figures 3.6, 3.8 and 3.10 are respectively two times, four times and eight times magnified images of "Lena" using

linear interpolation technique. Figures 3.12 and 3.13 are respectively the original and decoded images of a part of "LFA" image. Figures 3.20 and 3.21 are respectively the original and decoded images of a part of the "Seagull" image. Figures 3.28 and 3.29 are respectively the original and decoded images of a part of "Girl" image. The results of fractal based magnification of "LFA" image are shown in Figures 3.14, 3.16 and 3.18 while Figures 3.15, 3.17 and 3.19 are the results of magnified images of "LFA" using nearest niegbour technique. The results of the "Seagull" image used for magnification are shown in Figures 3.22, 3.24 and 3.26 while Figures 3.23, 3.25 and 3.27 are the results of magnified images of "Seagull" using nearest niegbour technique. The results of fractal based magnification of "Girl" image are shown in Figures 3.30, 3.32 and 3.34 while Figures 3.31, 3.33 and 3.35 are the results of magnified images of "Girl" using nearest niegbour technique. Magnified images of "LFA", "Seagull" and "Girl" obtained using linear interpolation have not been included.

From Table 3.1, it is evident that, in terms of the proposed error criterion, the performance of the proposed fractal based image magnification scheme is better than that of the linear interpolation technique as well as nearest neighbour technique. The results of linear interpolation are better than that of nearest neighbour method. The results presented in Table 3.2 show that, in terms of the similarity criterion, the nearest neighbour technique is a little better than that of the fractal based technique and that of the linear interpolation technique in some cases for magnification factor 2. But the the PIFS based technique appears to be better for a magnification factor more than two. On the other hand the present magnification method is appear to be better than that of the linear interpolation method in all cases. Comparing Figures 3.3, 3.4, and 3.5 with 3.7, 3.9, 3.11, 3.6, 3.8 and 3.10 visually, one can find some ringing and blurring are present in the case of nearest neighbour technique and linear interpolation technique for magnification of order more than two. On the other hand, in the case of proposed fractal based magnification, a few block effects have been observed. Similar conclusion is true for the results of the other images too.

3.7 Comparison With Other Fractal Based Magnification Techniques

So far only one attempt has been made towards image magnification using fractals. Polidori and Dugelay [141] suggested a zooming algorithm using fractal code. They pointed out the error in magnified images due to decoding of fractal code at any resolution. They tried to reduce the error by considering overlapped partitions of range blocks [150] while computing the fractal code. The given image was first subdivided into four parts and different overlapping partitioning schemes were adopted for different subdivisons. During encoding more than one estimates of a range block pixel were obtained due to overlapping range blocks. The final estimate was computed by averaging the estimates. The magnified image was then obtained by decoding the fractal code to the desired resolution. This scheme was also modified by considering the average of two closest estimates of a range block pixel. In both the cases, the best matched domain block was found by exhaustive search and by considering the maximum possible domain pool.

Huge computational time for finding fractal code is expected in the scheme described by Polidori and Dugelay [141]. The computational time is reduced significantly in our case by adopting GA based search. Moreover, while obtaining magnified images of higher resolution, no information regarding lower resolutions images was utilized by Polidori and Dugelay. This may incurred more magnification error as fractal coding is a lossy process. The proposed method has tried to reduce the magnification error of an image of higher resolution by minimizing the error at lower resolutions. Note that successive magnifications approach reduces the error at higher resolutions.

It has been reported by Polidori and Dugelay [141] that the results obtained by their scheme are no better than that obtained by linear interpolation method. The proposed method has been compared, in terms the distortion and similarity measures, with with the linear interpolation method (see Table 3.1 and Table 3.2). It has been observed that the results obtained by the present method are better than that obtained by linear interpolation technique. One may conclude that the performance of the proposed magnification technique is perhaps better than the technique of zooming using fractals by Polidori and Dugelay [141].

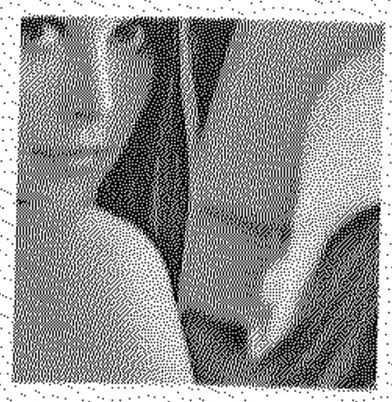


Fig. 3.1: A part of Lena image

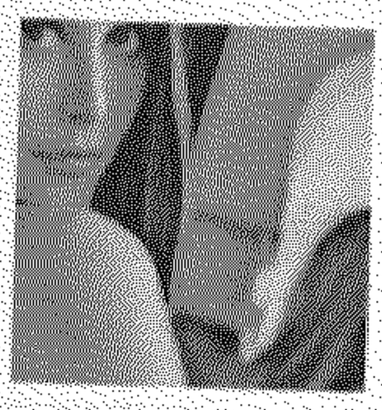


Fig. 3.2: Decoded image of part of Lena

Fig. 3.3: Two times magnified Lena using Fractal technique

Fig. 3.4: Four times magnified Lena using Fractal technique

Fig. 3.5: Eight times magnified Lena using Fractal technique

Fig. 3.6: Two times magnified Lena using Linear Interpolation

Fig. 3.7: Two times magnified Lena using Nearest Neighbour

3.8 Conclusions and Discussion

The most important advantage of the proposed technique of fractal image magnification is that it utilizes the coded (fractal) version of the input image instead of the original image. Therefore it is cost effective in the sense of storage space and time as no decoding is performed at the receiving end in case of transmission of the codes. The performance of the magnification algorithm is found to be satisfactory in terms of the proposed distortion measure and similarity criterion.

Another advantage of fractal image magnification is that it magnifies the image by expanding the fractal codes or the transformations which may be looked upon as independent of image resolution. The only error involved with it, is the problem of discretization. Thus the structure and the shape of the image remains almost same. In a sense, it is like approximation resulting in a sharper expanded image. Other image magnification schemes use pixel replication to expand image. Pixel replication makes an image blocky, blurry and patchy after a certain extent of expansion.

The proposed fractal image magnification algorithm is implemented with magnification factors which are powers of 2. But in practice one may need to magnify the given image by other factors too. One way of performing the magnification of order k is to select the domain blocks having number of rows and number of columns k times larger than that of the range block. Then find the mapping of domain block to range block by spatial averaging of the pixel values of the domain block and fitting a straight line (least square

Fig. 3.8: Four times magnified Lena using Linear Interpolation

Fig. 3.9: Four times magnified Lena using Nearest Neighbour

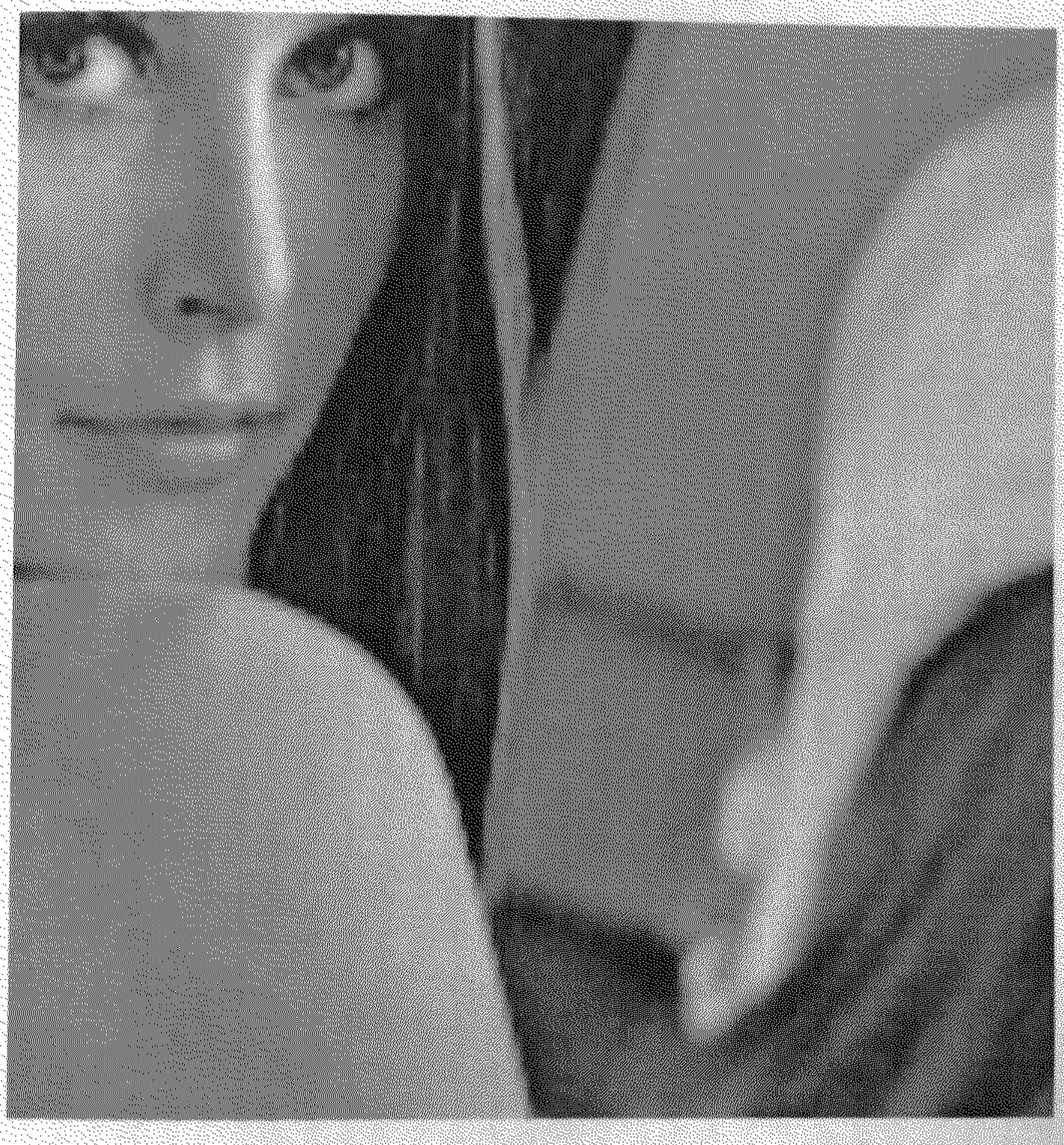


Fig. 3.10: Eight times magnified Lena using Linear Interpolation

Fig. 3.11: Eight times magnified Lena using Nearest Neighbour

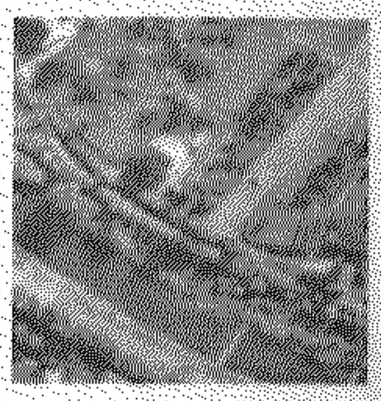


Fig. 3.12: A part of LFA image

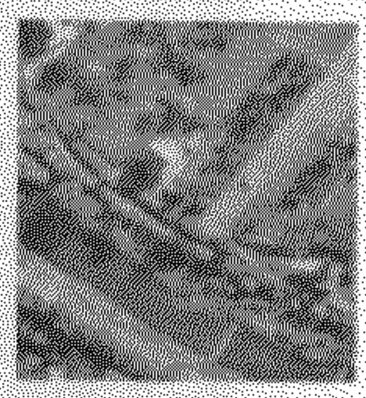


Fig. 3.13: Decoded image of part of LFA image

Fig. 3.14: Two times magnified LFA using Fractal technique

Fig. 3.15: Two times magnified LFA using Nearest Neighbour technique

Fig. 3.16: Four times magnified LFA using Fractal technique

Fig. 3.17: Four times magnified LFA using Nearest Neighbour technique

Fig. 3.18: Eight times magnified LFA using Fractal technique

Fig. 3.19: Eight times magnified LFA using Nearest Neighbour

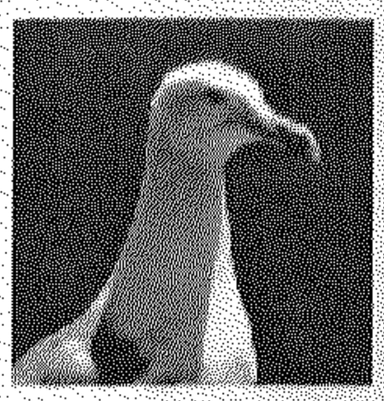


Fig. 3.20: A part of Seagull image

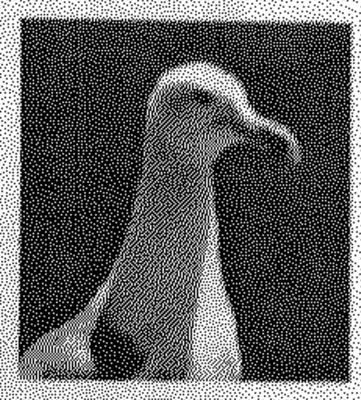


Fig. 3.21: Decoded image of part of Seagull image

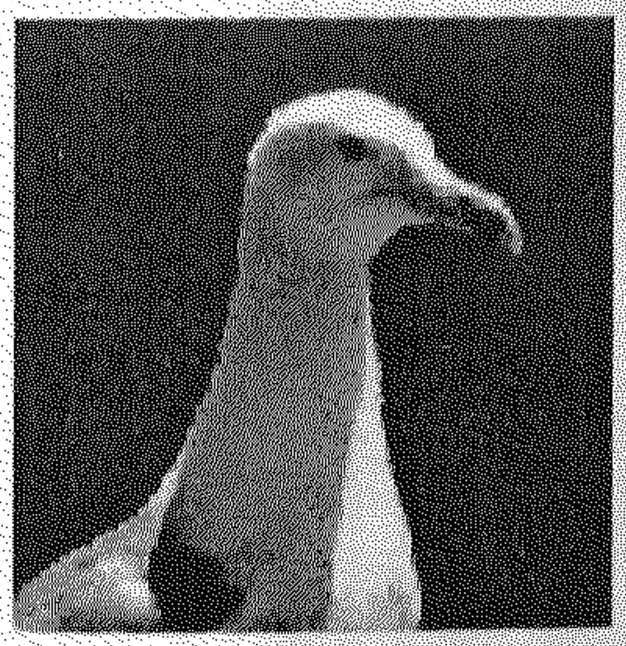


Fig. 3.22: Two times magnified Seagull using Fractal technique

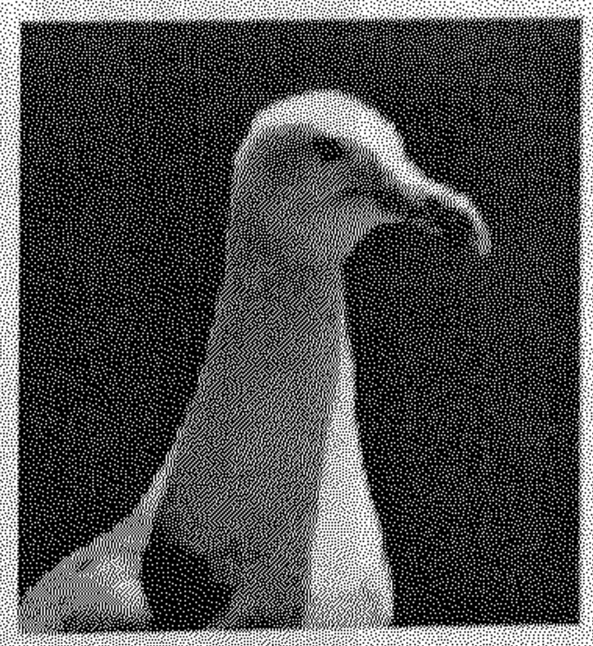


Fig. 3.23: Two times magnified Seagull using Nearest Neighbour technique

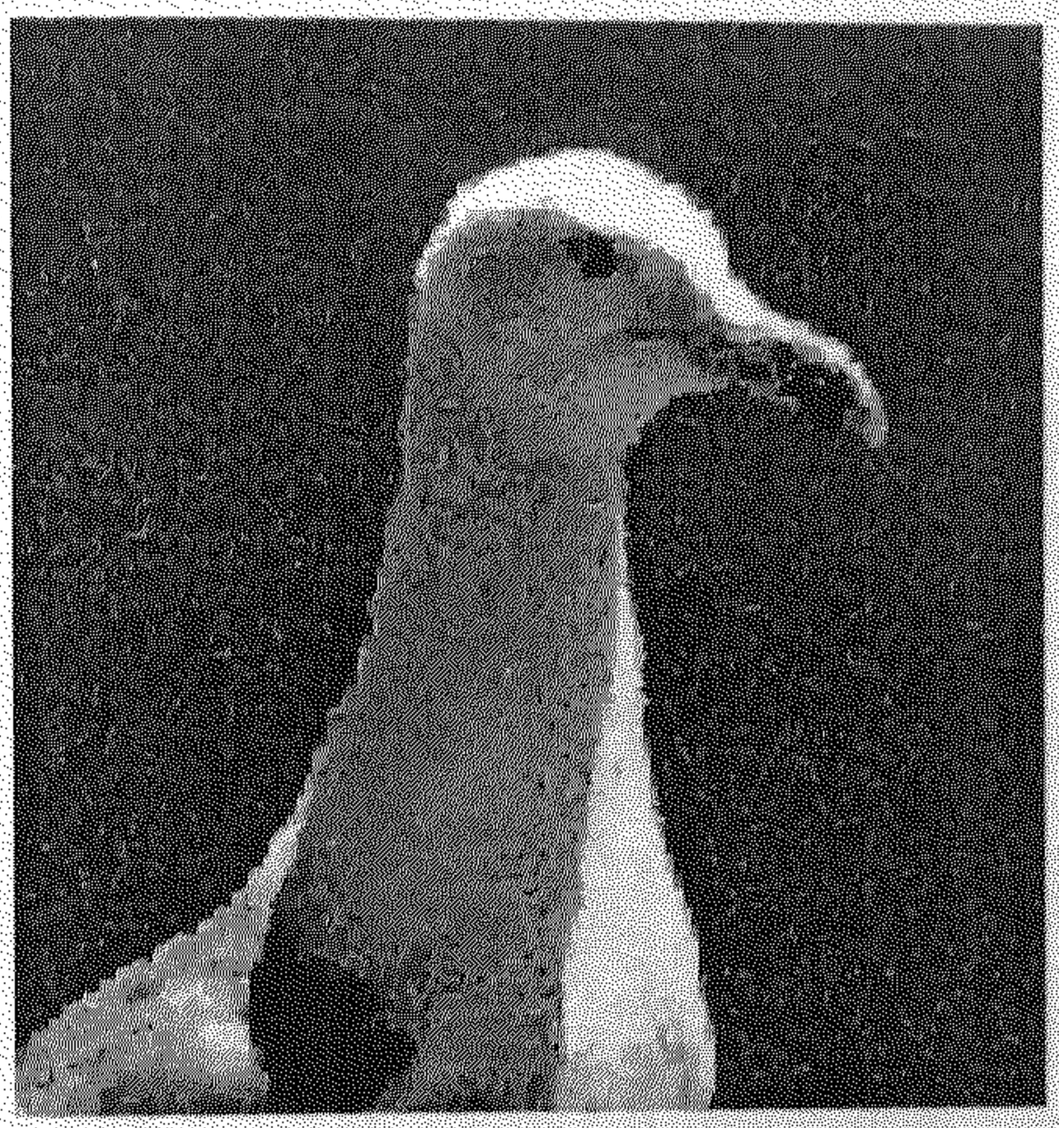


Fig. 3.24: Four times magnified Seagull using Fractal technique

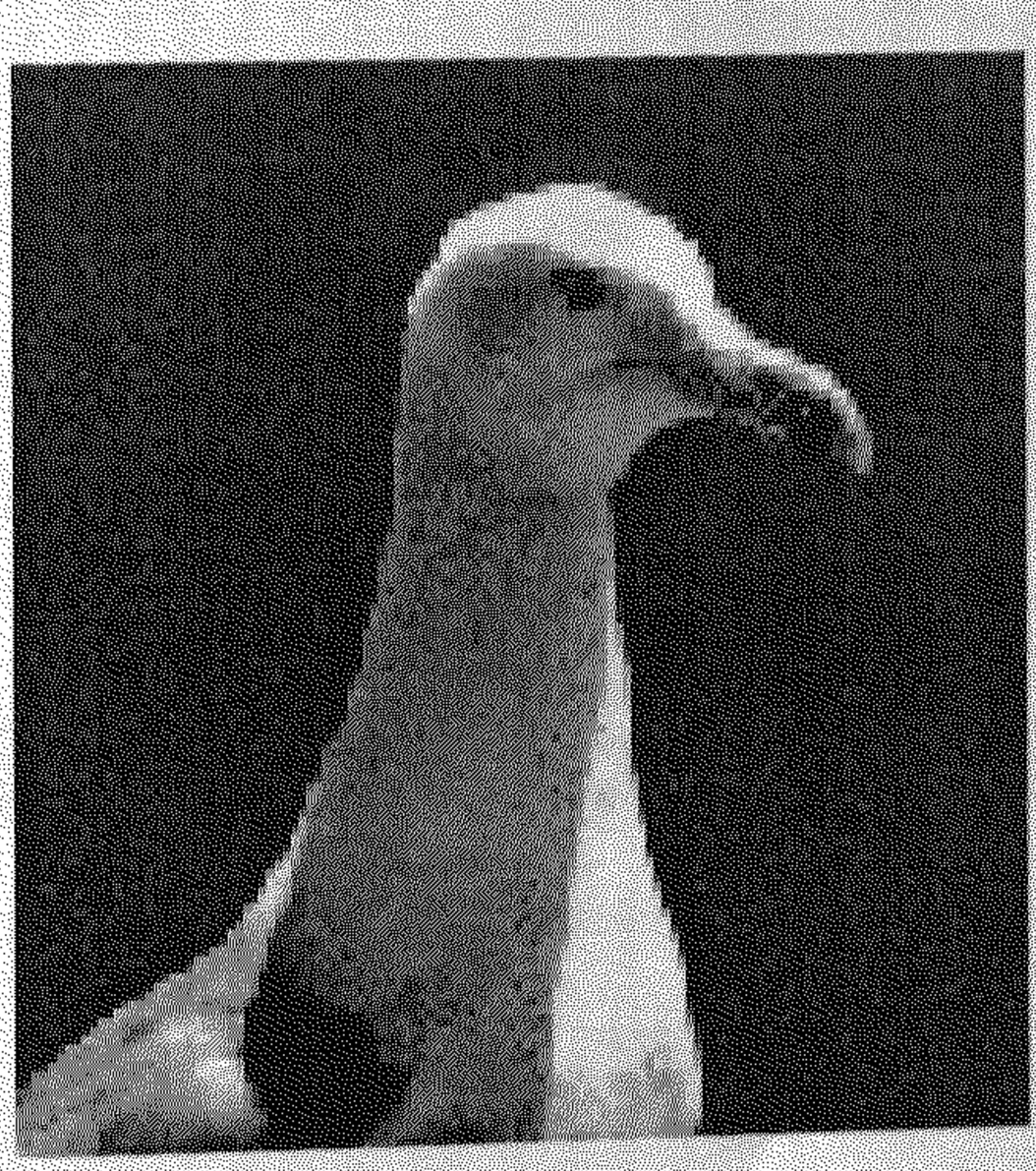


Fig. 3.25: Four times magnified Seagull using Nearest Neighbour technique

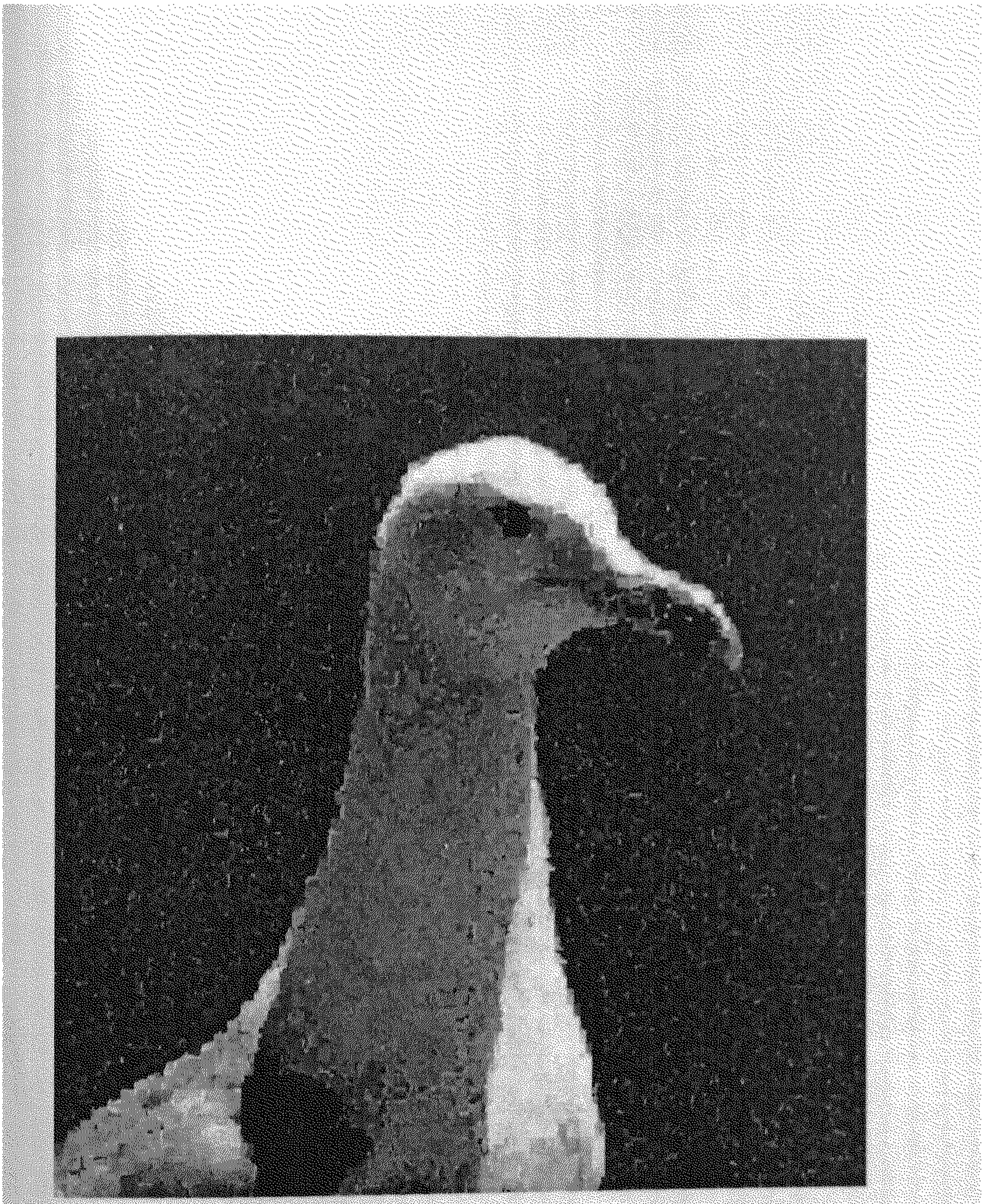


Fig. 3.26: Eight times magnified Seagull using Fractal technique

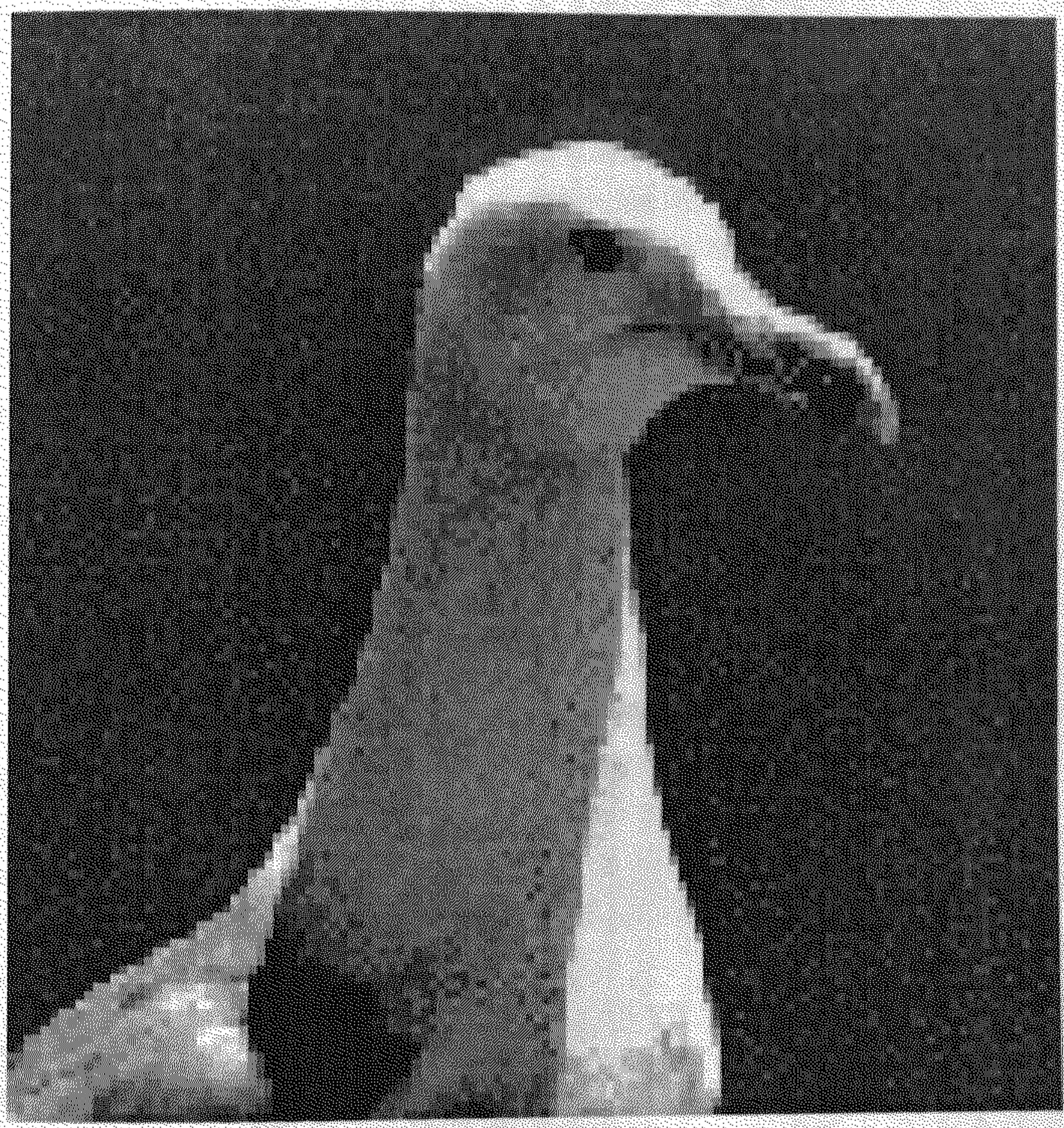


Fig. 3.27: Eight times magnified Seagull using Nearest Neighbour

analysis). But if k is large *i.e.* if the size of the domain block is very large compared to that of range block, the similarity patterns between range blocks and domain blocks may not be observed. In that case, true magnification will not be possible. To avoid this, magnification task can be divided into several steps as described in Section 3.3.2.

Another way of performing magnification by a factor k is as follows. Magnified image can also be achieved by considering the normalized distance between the range blocks and their matched domain blocks in the original image. The normalized distance will only keep track of domain block location in a magnified image irrespective of its resolution. The normalized distances are stored in the PIFS codes for each range block. Now, to achieve magnification of order k from PIFS codes, the location of the matched domain blocks for range blocks may be obtained using the normalized distances and k. Once the matched domain block is fixed, the rest is similar to magnification by factor 2, *i.e.*, mapping from domain block to range block is computed by taking spatial averaging of domain block pixel and using least square analysis. While performing the magnification task, if the distance between the range block and the domain block appears to be fractional, one has to discretize the distance. The problem of k being large can be handled by dividing the magnification task into several steps and by modifying the PIFS codes in each step.

The size of the range block plays a vital role in fractal image compression and fractal image magnification. In particular these two algorithms are in opposite direction from the point of view of range block size. So, one can think of an optimal range block size for which good quality magnified images can be reconstructed from the fractal codes and at the same time considerable amount of compression (in terms of compression ratio) can be achieved.

A new distortion measure or fidelity criterion to judge the performance of the proposed algorithm have been introduced. There are other available methods such as non-parametric statistical tests [148] for the same purpose. The common tests for examining the degree of association between two distributions whose distribution functions are unknown are Sign test, Wald Wolfowitz Run test, Wilcoxson test and Kolmogorov Smirnov test. Another evaluating criteria based on fractal dimension has been suggested by Lalitha et al. [95]. But the most important feature which should be considered while examining the distortion between two images is the edge distribution of the images as the edges are very sensitive to human eyes. Neither the fractal based

Fig. 3.28: A part of Girl image

Fig. 3.29: Decoded image of part of Girl image

evaluating criteria nor the statistical tests take care of distortions present in the edges. The main advantage of the proposed error criterion is that it takes care of distortions present in the edge images. Thus, one of the important tasks is to find the proper edges in the images for the implementation of the proposed distortion measure. A very simple technique for the detection of edges has been used here, though one may suggest more complex techniques for it.

The other measure proposed for judging the performance of the proposed fractal image magnification technique is JND based similarity criterion. This measure also takes care of the distribution of edges as JND is basically the change in contrast of an object with respect to its background. But one disadvantage of this similarity measure is that it deals with the change in pixel values ignoring the edge pattern.

In the next chapter we are going to describe a new PIFS based image edge extraction technique.

Fig. 3.30: Two times magnified Girl using Fractal technique

Fig. 3.31: Two times magnified Girl using Nearest Neighbour technique

Fig. 3.32: Four times magnified Girl using Fractal technique

Fig. 3.33: Four times magnified Girl using Nearest Neighbour technique

Fig. 3.34: Eight times magnified Girl using Fractal technique

Fig. 3.35: Eight times magnified Girl using Nearest Neighbour

Chapter 4

Image Edge Extraction in the Compressed Domain

4.1 Introduction

The storage of large volume of images in digital library usually requires a great deal of compression. Thus, instead of performing image processing tasks using digital images, one may be interested to perform the same task in the compressed domain. By doing so one can save a lot of computer time. To facilitate this, an edge extraction scheme using fractal reconstruction is proposed in this chapter. In particular, the proposed scheme is embedded in the decoding process of PIFS code with a restriction that the reconstruction process should start with a blank image.

Edge extraction plays a very important role in many image processing applications. Edges in an image are variations of some physical properties, like surface illumination and shadows, geometry and reflectance of objects in the scene [30]. Usually, edges in an image are formed due to changes or discontinuities in image intensity. Hence one can say that edge points represent some features of an image. The process of extraction of these features is called edge extraction. It can also be viewed as extracting features in pattern recognition problems where the given image can be assumed to be a target pattern. In the process of edge extraction, an edge image is produced from a gray level image. A great deal of effort has been directed towards solving this problem [28, 30, 68, 93, 110, 145], yet complete success is not achieved as edge semantics are extremely complicated. Our understanding of proper edge is still inadequate as there is hardly any measure to judge the true edge in an image. Most of the common algorithms use kernels known as edge operators for finding edges in an image. These kernels are based on local gradients typically computed over a small window which scan the whole image. Then, from the gradient information, edges are computed using thresholding approach [28, 68, 110].

Digital images used for analysis in modern image processing are usually of the form of a two dimensional array of pixels, presented with a constant number of bits per pixel. Commonly, the images are accessed for analysis from the image-data-base where images are stored in compressed form. The storage of large volume of data in applications like remote sensing imagery and moving video image sequence requires great deal of compression. The transmission of huge data through narrow band width channel also requires image compression. Therefore, instead of analyzing the raw digital images, one may be interested to develop methods for analysis from the coded representation of the

image. The advantages of developing algorithms which utilize the coded version are two fold. Firstly it will reduce the storage requirements. Secondly it may appear to be cost effective in the sense of computational time as no prior reconstruction of the image is needed in the image processing task. Similarly, in the case of transmission, if the coded version of the image is directly utilized at the receiving end for subsequent analysis then considerable amount of computer time may be saved. So far only a few attempts have been made to utilize the coded version of the images for performing image edge detection task [111]. In the present chapter, a new edge extraction technique which can be applied directly on the coded version of the image is developed. In particular, we have used fractal code of the image instead of the original one for performing edge extraction.

Several algorithms with different motivations have been suggested in the literature to obtain PIFS or fractal code for a given image. Detailed description of these techniques are available in Chapter 1. In the present case, to extract edges, PIFS code as described in Chapter 2 has been used. The proposed scheme is a part of the decoding scheme of PIFS code which results in an image which is an edge image of the original image. Actually, the proposed edge extraction scheme is embedded in the decoding process of PIFS code. The scheme is unique of its kind as it is not using any kind of convolution operation based on kernels, which is very common in conventional edge detection algorithms.

The methodology of image encoding and decoding using PIFS technique are outlined in Section 4.2 and Section 4.3 respectively. The methodology of using fractal code for edge extraction of a given image using PIFS is described in Section 4.4. Section 4.5 presents implementation and the results. Discussion and conclusions are provided in Section 4.6.

4.2 Brief description of the Construction of PIFS Code

Let I be a given digital image having size $z \times z$ and the range of gray level values be $\{0, 1, 2, \cdots, l-1\}$. Thus the given image I can be expressed as a matrix $((e(i,j)))_{z\times z}$, where i and j stand for row number and column number respectively and $e(i,j)$ rep-

resents the gray level value for the position (i, j). The image is partitioned into n non overlapping squares of size, say $b \times b$, and let this partition be represented by $\mathcal{N} = \{\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_n\}$. Each $\mathcal{R}_i$ is named as range block. Note that the number of range blocks $n = \frac{z}{b} \times \frac{z}{b}$. Let $\mathcal{M}$ be the collection of all possible blocks of size $2b \times 2b$ in the image. Let $\mathcal{M} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m\}$. Here $m = (z - 2b + 1) \times (z - 2b + 1)$ and $\mathcal{D}_j$'s are named as "domain blocks".

Now, let us define,

$$\mathcal{A} \;=\; \{1, 2, \cdots, z\} \times \{1, 2, \cdots, z\} \times \{0, 1, 2, \cdots, l-1\}.$$

Here $\mathcal{A} \subset \mathbb{R}^3$. Note that any image I is a subset of $\mathcal{A}$ but any subset of $\mathcal{A}$ is not necessarily an image. Also $\mathcal{R}_i \subset \mathcal{A} \;\; \forall i$ and $\mathcal{D}_j \subset \mathcal{A} \;\; \forall j$.

Let, for a range block $\mathcal{R}_i$,

$$\mathcal{F}_j = \{f : \mathcal{D}_j \to \mathcal{A} \;\; ; \;\; f \;\text{ is an affine contractive map}\}.$$

Let, $f_{i|j} \in \mathcal{F}_j$ be such that

$$\rho(\mathcal{R}_i,\; f_{i|j}(\mathcal{D}_j)) \;\leq\; \rho(\mathcal{R}_i,\; f(\mathcal{D}_j)) \;\; \forall f \in \mathcal{F}_j,\; \forall j.$$

Here ρ is a suitably chosen distance measure. Note that, in this thesis, we have considered 'ρ' to be RMSE.

Now let k be such that

$$\rho(\mathcal{R}_i,\; f_{i|k}(\mathcal{D}_k)) \;=\; \min_j\{\; \rho(\mathcal{R}_i,\; f_{i|j}(\mathcal{D}_j))\; \}. \tag{4.1}$$

Also, let $\quad f_{i|k}(\mathcal{D}_k) = \widehat{\mathcal{R}}_{i|k}.$

The aim here is to find $f_{i|k}(\mathcal{D}_k)$ for each $i \in \{1, 2, \cdots, n\}$. The set $\mathcal{K} \;=\; \{\mathcal{K}_i,\; i = 1, 2, \cdots, n\}$ is called the obtained fractal code. The mapping of range blocks from matched domain blocks are shown in Figure 4.1.

The affine contractive transformation $f_{i|j}$ is constructed using the fact that the gray values of the range block are a scaled, translated and rotated version of the gray values of domain block. The contractive affine transformation $f_{i|j}$ is defined in such a way that $f_{i|j}(\mathcal{D}_j)$ is close to $\mathcal{R}_i$. Also $f_{i|j}$ can be separated into two parts, one for spatial information and the other for information of gray values. The first part indicates which

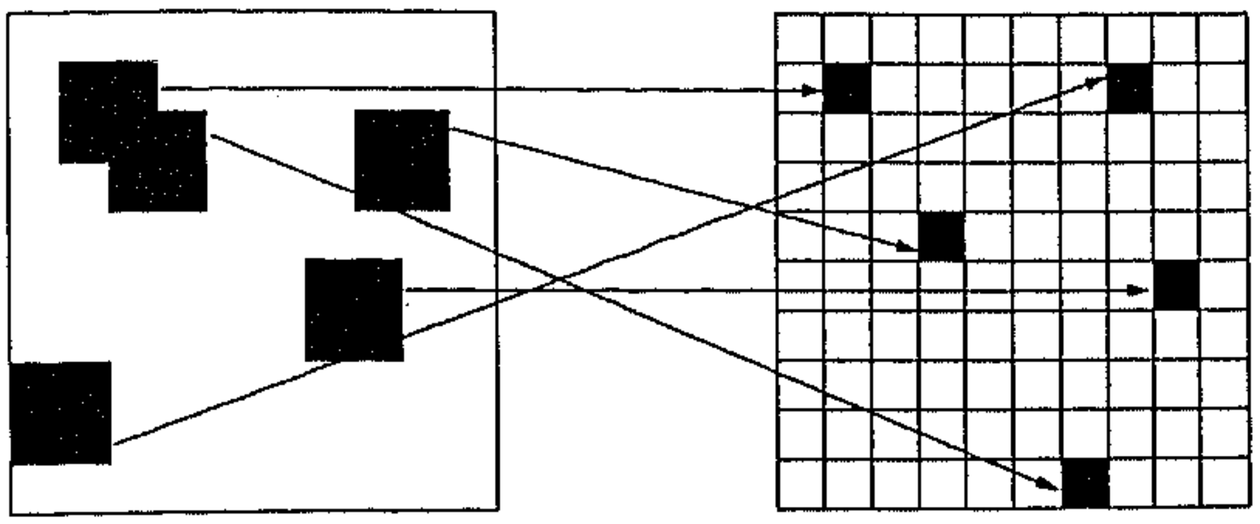


Fig. 4.1: Mapping from domain blocks to range blocks in PIFS scheme

pixel of the domain block corresponds to which pixel of range block. The second part expresses the scale and shift parameters for the set of pixels of the domain block to the range block.

The first part is the shuffling of the pixel positions of the domain block. Second part is the estimation of a set of values (gray values) of range blocks from the set of values of the matched domain blocks. These estimates are obtained by using the least square analysis of ordered pairs of values (See Chapter 2).

The size of a matched domain block is double that of any range block. But, the method of least squares (straight line fitting) needs point to point correspondence (see Chapter 2). To overcome this, one has to construct contracted domain block such that the number of pixels in a contracted domain block becomes equal to that of any range block. The contracted domain blocks are obtained by adopting any one of the following two techniques. In the first technique, as shown in Figure 4.2, for a 4×4 domain block, the average values (integers) of four pixel values in 2×2 non overlapping squares within the domain block are considered as the pixel values of the contracted domain block. In this scheme, row number and column number corresponding to each pixel value of the contracted domain block are equal to the row number and column number of the topmost pixel value in every 2×2 square considered within the domain block [80]. In the other scheme, as shown in Figure 4.3, for a 4×4 domain block, contracted domain block is constructed by taking pixel values along with the row number and column

number from every alternative rows and columns of the domain block. The first one is termed as spatial averaging and the second one is named as subsampling.

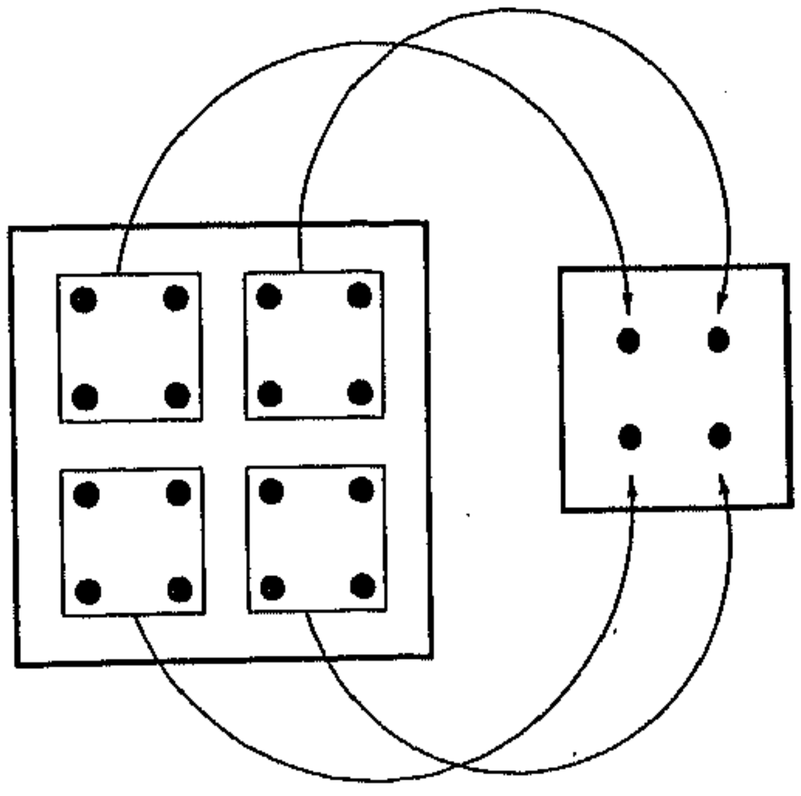


Fig. 4.2: Construction of contracted domain block : Scheme 1

Now to select an appropriately matched contracted domain block ($\mathcal{D}_k$) and appropriately matched transformation ($f_{i|k}$) for a range block ($\mathcal{R}_i$), the distance measure "ρ" [used in equation (4.1)] is taken to be the simple Root Mean Square Error (RMSE) between the original set of gray values and the obtained set of gray values of the concerned range block.

4.3 Image Reconstruction Using PIFS

The detailed mathematical description of the IFS theory and PIFS theory are available in Chapters 1 and 2. The edge extraction procedure proposed in the present chapter is actually embedded in the reconstruction procedure of the fractal codes. The following is the description of the reconstruction process of the fractal code. The same notations as used in Chapter 3, are used here.

Let us consider the generated PIFS code $\mathcal{K} = \{\mathcal{K}_i : i = 1, 2, \cdots, n\}$, where $\mathcal{K}_i = \{\mathcal{D}_k, f_{i|k}\}$, of a given image I. Here, $\mathcal{D}_k$ is the location of matched domain block and $f_{i|k}$ is the appropriate transformation for range block $\mathcal{R}_i$. The fractal code $\mathcal{K}$

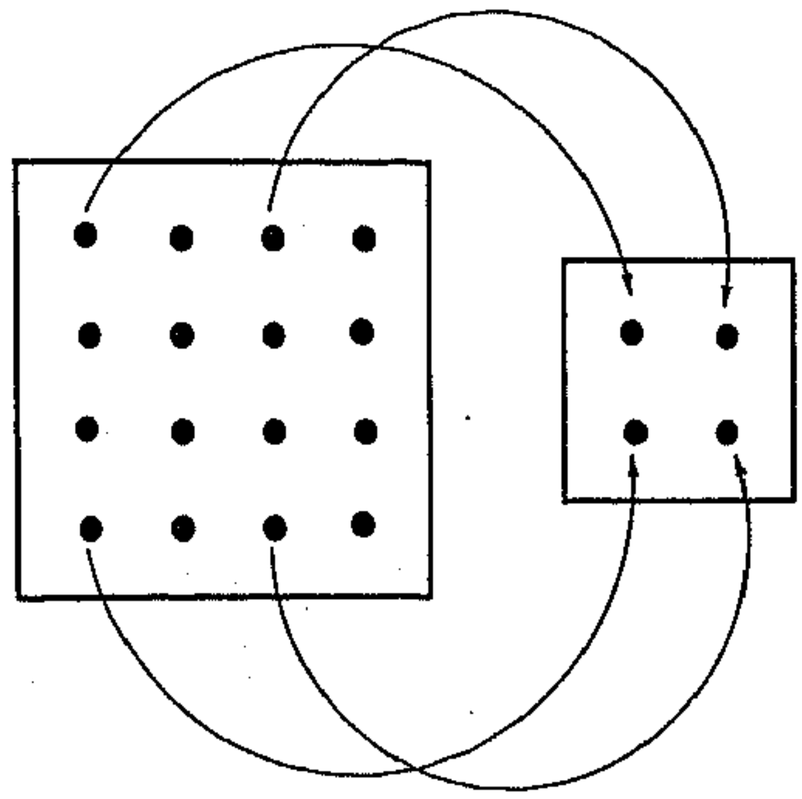


Fig. 4.3: Construction of contracted domain block : Scheme 2

is obtained through the coding procedure described in Chapter 2. The natural decoding scheme simply consists in iterating the code $\mathcal{K}$ on any initial image, say I_0, until convergence to a stable decoded image is observed. Thus, the decoding process is of the form $\{\mathcal{K}^N(I_0)\}_{N>0}$. At the Nth iteration, the image I_N is used as input to the decoding system, where I_N is the output image obtained from the $(N-1)th$ iteration. The mapping of an image under the fractal code is done sequentially. In particular for each index i, the transformation $f_{i|k}$ is applied to the domain block $\mathcal{D}_k$ of the current input image, and mapped on to the range block $\mathcal{R}_i$ of the current output image which is input image of the next iteration. The sequence of images $\{I_N\}_{N>0}$, is called the reconstruction sequence for the code $\mathcal{K}$ with a starting image I_0. Figure 4.4 shows output images of first three and last four iterations of fractal code of "Lena" image, with the starting image a "Rose" image. In this case we have stopped the process of decoding after ten iterations.

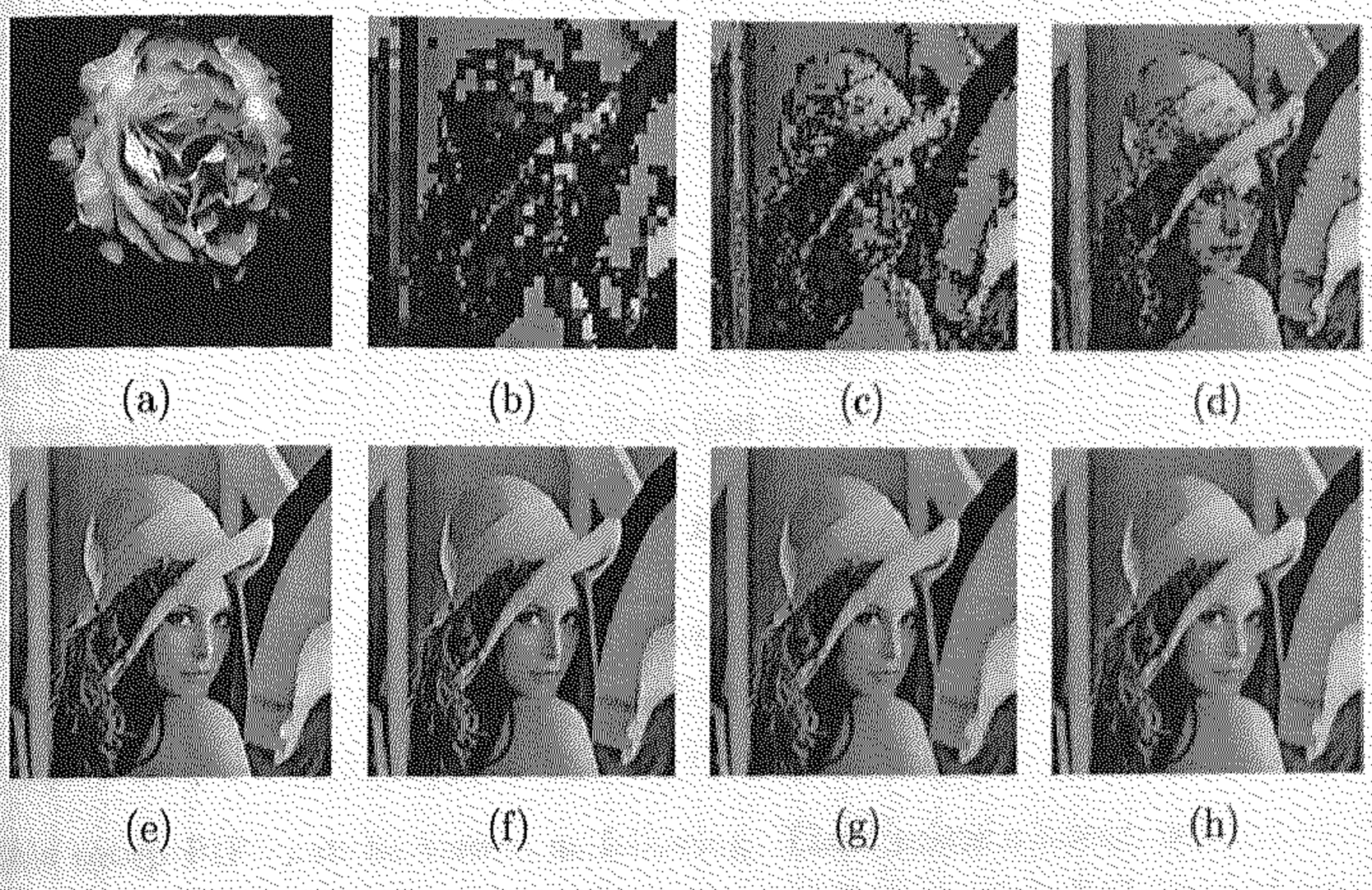


Fig. 4.4: Decoded images of Lena after 1st (b), 2nd (c), 3rd (d), 7th (e), 8th (f), 9th (g) and 10th (h) iteration with starting image Rose (a)

4.4 Methodology of Edge Extraction from Intermediate Representation

Edge extraction using PIFS code is a sequential process. In this process, edges are extracted from the image decoding process (described in Section 4.3). In other words, a decoding process is carried out with some modifications. Now for the better understanding of the process and to know the motivation behind such an algorithm one should know the convergence procedure of each pixel of a given image under PIFS setup. By the convergence of each pixel we mean to say that how pixels of the given image converge to fixed point through an iterative sequence. Following section provides a mathematical formulation to show the existence of the fixed point of PIFS at pixel level. This study is also going to show that different pixels take different time to converge. In the present case, the technique of subsampling (Section 4.2 is used to show the computation of fixed points of pixels. Note that, the case of averaging is very difficult to prove in the same way as the computational complexity will be huge to handle, though the same conclusion, regarding the time of convergence of a pixel, is

expected.

4.4.1 Convergence of PIFS

Let, I be a given image having size $z \times z$ and the range of gray level values be $\{0, 1, 2, \cdots, l-1\}$. For this given image we can construct a vector $\underline{x}$ whose elements are the pixel values of the given image I. Note that there are z^2 pixel values of I. Thus,

$$\underline{x} = (x_1, \quad x_2, \quad x_3, \quad \cdots \quad , x_{z^2})'$$

is the given image where x_1 is the pixel value corresponding to the $(1,1)th$ position of I. Likewise, for a pixel value x_r corresponding to the $(i,j)th$ position of I, the value of r is, $r = (i-1)\,z + j, \quad 1 \leq i, \; j \leq z$.

In this setup PIFS can be viewed as following. There exists a linear, not necessarily strictly contractive, map for each element of $\underline{x}$ and this map is called forward map of the element. In the process of iteration, the input to a forward map will be any one of the z^2 elements of $\underline{x}$ and the map is called backward map for this input element. Thus, if a map f_i (say) is used to estimate the element x_i from the element x_j, then the map f_i is called forward map of x_i and backward map of x_j. Thus for each element of $\underline{x}$ there exists a forward map. On the other hand, an element of $\underline{x}$ can have one or more or no backward map(s). The set $\mathcal{F}$, of forward maps, is called the PIFS codes of I.

Let us define a set $\mathcal{Q}$ as

$$\mathcal{Q} = \{0, 1, 2, \cdots, l-1\}$$

Now consider the set $\mathcal{S}$ where,

$$\mathcal{S} = \{\, \underline{x} \mid \underline{x} = (x_1, \quad x_2, \quad x_3, \quad \cdots \quad , x_{z^2})', \quad x_i \in \mathcal{Q} \,\}.$$

$\mathcal{S}$ is the set of all possible images. The given image I is surely an element of $\mathcal{S}$, *i.e.*, $I \in \mathcal{S}$. The PIFS codes $\mathcal{F}$ can be looked upon as

$$\mathcal{F} \; : \; \mathcal{S} \; \rightarrow \; \mathcal{S} \; .$$

The attractor of $\mathcal{F}$, $\underline{a}$ (say), if exists, will also belong to $\mathcal{S}$. So, the first task is to show the existence of $\underline{a}$.

Let f_1 be the forward map for a particular element x_{r_1}, where $r_1 = (i_1-1)z+j_1$. Also let this element be mapped from the element x_{r_2} ($r_2 = (i_2 - 1)\, z + j_2$). Thus f_1 is the backward map for x_{r_2}. Again x_{r_2} is being mapped from x_{r_3}, $(r_3 = (i_3-1)z+j_3)$ with a forward map f_2. Thus we have a sequence of maps for the element x_{r_1} as following.

$$(i_1, j_1) \xleftarrow{f_1} (i_2, j_2) \xleftarrow{f_2} (i_3, j_3) \xleftarrow{f_3} \cdots \xleftarrow{f_{m-1}} (i_m, j_m);\ m \leq (z^2 - 1)\,. \quad (4.2)$$

The above sequence will be stopped at (i_m, j_m) if

$$(i_{m+1}, j_{m+1}) = (i_k, j_k);\ \text{for } k = 0 \text{ or } 1 \text{ or } 2 \text{ or } \cdots \text{ or } m. \quad (4.3)$$

The stopping phenomenon of this sequence is mandatory as there are finite number (z^2) of elements in $\underline{x}$. Moreover each elements of $\underline{x}$ possesses same type of sequence in PIFS codes. Thus it is enough to show that the element x_{r_1} has got a fixed point in the process of iteration and this will lead to prove the existence of $\underline{a}$ (attractor of $\underline{x}$).

It is clear from the sequence (4.2) that during the iterative process the element x_{r_1} will have a fixed point once the element x_{r_2} is fixed. Again the convergence (to a fixed point) of the element x_{r_3} confirms the convergence of the element x_{r_2} and likewise for the rest of the elements. Thus convergence of the last element of the sequence implies the convergence of the rest of the elements. The convergence of the last element of the sequence is possible in four different ways according to the stopping condition (4.3).

An important point to be noted in this context is the problem of discretization. To get the decoded image in an iterative process using PIFS codes, one needs to discretize the output. This can be done in two ways. One is the discretization of the output in each iteration. Another is discretization at the end of the iterative process. The iterative process is stopped whenever there is no change in gray values in two successive iterations. To prove the convergence of the elements in four different ways we have used the discretization of the second type.

Case 1 : $m = 1$.
Here $(i_2, j_2) = (i_1, j_1)$.

It implies that (i_1, j_1) is mapped into itself with a map f_1.
Here $f_1 = a_1 x + b_1$; $x \in Q$ and $0 \leq a_1 < 1$.
Note that in this case the affine map f_1 should necessarily be a strictly contractive map otherwise the element will not converge to a fixed point.

If we start with any value ($x \in Q$) of (i_1, j_1), the element will converge to the fixed point $\frac{b_1}{1 - a_1}$.

Case 2 : $m > 0$ and $k = m$
Here $(i_{m+1}, j_{m+1}) = (i_m, j_m)$.
It implies that (i_m, j_m) is mapped into itself with a map $f_m = a_m x + b_m$; $x \in Q$ and $0 \leq a_m < 1$. Thus (i_m, j_m) will converge to $\frac{b_m}{1-a_m}$. Once (i_m, j_m) is fixed at $\frac{b_m}{1-a_m}$, the element (i_{m-1}, j_{m-1}) will be fixed at

$$\frac{a_{m-1}\, b_m}{1 - a_m} + b_{m-1} = \frac{a_{m-1}\, b_m - a_m\, b_{m-1} + b_{m-1}}{1 - a_m}.$$

In this case the forward map is $f_{m-1} = a_{m-1} x + b_{m-1}$; $0 \leq x < l$. Again (i_{m-1}, j_{m-1}) is fixed implies convergence of (i_{m-2}, j_{m-2}) with forward map $f_{m-2} = a_{m-2} x + b_{m-2}$; $x \in Q$, at

$$\frac{a_{m-2}\, a_{m-1}\, b_m - a_{m-2}\, a_m\, b_{m-1} + a_{m-2}\, b_{m-1} - a_m\, b_{m-2} + b_{m-2}}{1 - a_m}.$$

Proceeding in this way, the fixed point of (i_1, j_1) is found out to be

$$\frac{a_1\, a_2 \ldots a_{m-1}\, b_m + (a_1\, a_2 \ldots a_{m-2}\, b_{m-1} + a_1\, a_2 \ldots a_{m-3}\, b_{m-2} + \ldots + a_1\, a_2\, b_3 + a_1\, b_2 + b_1)(1 - a_m)}{1 - a_m}.$$

Note that in this case the affine map f_m should necessarily be contractive in strict sense. But the rest of the maps need not be strictly contractive. The eventual contractivity associated with the element $x_{r_1} = (i_1, j_1)$, will be $s_{r_1} = \prod_{i=1}^{m} a_i$.

Case 3 : $m > 0$ and $k = 1$
Here $(i_{m+1}, j_{m+1}) = (i_1, j_1)$.
It implies that the starting and the last element of the sequence (4.2) is same. This can be looked as a complete loop for the sequence. This case has been solved stepwise. First of all the case is solved for $m = 2$, and $m = 3$. Then on the basis of these

the fixed point for the case of general m is solved. Note that, in a complete loop, each element in the loop attains fixed point simultaneously.

Case 3(a) : $m = 2$

Here we have only two elements viz. (i_1, j_1) and (i_2, j_2). The element (i_1, j_1) is being mapped from the element (i_2, j_2) by the affine map $f_1 = a_1 x + b_1 ; x \in Q$. On the other hand the element (i_2, j_2) is being mapped from (i_1, j_1) by the affine map $f_2 = a_2 x + b_2 ; x \in Q$.

$$(i_1, j_1) \xleftarrow{f_1} (i_2, j_2) \xleftarrow{f_2} (i_1, j_1).$$

Let x be the starting value of (i_1, j_1) and y be the starting value of (i_2, j_2). After first iteration the values of (i_1, j_1) and (i_2, j_2) will be $a_1 y + b_1$ and $a_2 x + b_2$ respectively. Again after second iteration these will be $a_1 a_2 x + a_1 b_2 + b_1$ and $a_2 a_1 y + a_2 b_1 + b_2$ respectively. Proceeding this way after infinite (practically large but finite) number of iterations, the fixed point of (i_1 , j_1) and (i_2 , j_2) will be independent of x and y. The Coefficients of x and y after N (even) iterations will be $(a_1 a_2)^{N/2}$ which tends to zero as N tends to infinity. The fixed points of (i_1, j_1) thus will be

$$\frac{a_1 b_2 + b_1}{1 - a_1 a_2}.$$

The same for the element (i_2, j_2) will be

$$\frac{a_2 b_1 + b_2}{1 - a_1 a_2}.$$

Note that both the maps need not be contractive. Moreover the eventual contractivity associated with the element x_{r_1} is $|a_1 a_2|$ which should be less than one.

Case 3(b) : $m = 3$

Here we have three elements viz. (i_1, j_1), (i_2, j_2) and (i_3, j_3). These three elements are making a complete loop in the sequence. The sequence of forward and backward maps is as follows;

$$(i_1,\ j_1) \xleftarrow{f_1} (i_2,\ j_2) \xleftarrow{f_2} (i_3,\ j_3) \xleftarrow{f_3} (i_1,\ j_1).$$

Taking the starting values of three elements as x, y and z and proceeding as *case 3(a)* we have the following results.

The fixed point of $(i_1,\ j_1)$ will be

$$\frac{a_1 (a_2 b_3 + b_2) + b_1}{1 - a_1 a_2 a_3}.$$

The element $(i_2,\ j_2)$ will converge to

$$\frac{a_2 (a_3 b_1 + b_3) + b_2}{1 - a_1 a_2 a_3}.$$

The fixed point of $(i_3,\ j_3)$ will be

$$\frac{a_3 (a_1 b_2 + b_1) + b_3}{1 - a_1 a_2 a_3}.$$

Here also the maps need not be contractive in the strict sense. The eventual contractivity will be $a_1 a_2 a_3$ in this case.

Case 3(c) : General m

Here we have m elements which are making a complete loop of sequence. It is clear from case *3(a)* and *case 3(b)* that all the elements of this sequence will have a fixed point after a large but finite number of iteration. Also the affine maps which are used, need not to be contractive. In particular, in this case the element $(i_1,\ j_1)$ will converge to

$$\frac{a_1 (a_2 (\ \dots\ (a_{m-2} (a_{m-1} b_m + b_{m-1}) + b_{m-2}) + \ \dots\ ) + b_2) + b_1}{1 - a_1 a_2 \ \dots\ a_m}.$$

Also the eventual contractivity for the element is $s_{r_1} = \prod_{i=1}^{m} a_i$.

Case 4 : $m > 0$ and $0 < k < m$

Here $(i_{m+1},\ j_{m+1}) = (i_k,\ j_k)$, where $k = 2$ or 3 or $\dots$ or $m-2$.

Without loss of generality say, $1 < k = m_0 < m-1$.
This case can be viewed as mixture two cases. Taking (i_{m_0}, j_{m_0}) as the starting element, a complete loop of sequence can be formed with rest of the elements. Thus, one can find the fixed point of this element as it is nothing but *case 3*. Once the element (i_{m_0}, j_{m_0}) is fixed then the fixed point of the original starting element (i_1, j_1) can be found out by using *case 2*. Like all the previous cases the eventual contractivity, in this case, will be

$$s_{r_1} = \prod_{i=1}^{m} a_i .$$

The above stated four cases provide the fixed points of pixels of a given image under PIFS setup. Note that for each pixel there will be a sequence of the form (4.2) and each pixel has fixed points. This sequence will follow any one of the above mentioned four cases. The contractivity factor associated with each pixel will be the product of all the scaled parameters (a_i) of the forward maps.

4.4.2 Construction of Edge Image using PIFS

So far, we were able to establish that, by construction, PIFS code is such that each pixel of the given image has a sequence of the form (4.2). Let us consider the number of elements included in a sequence as the length of the sequence. So, from (4.2) it is quite evident that different elements will have different sequences with different lengths. Now, since each of the elements has a fixed point, the time taken to converge will be different for different elements. In other words, different elements converge to their respective fixed points at different iterations. The most common question that arises at this juncture is : why the convergence rate is different for different elements. To answer this question, one should recall the construction procedure of the PIFS code. Note that, for the construction of the PIFS code, a range block could be either smooth or rough. PIFS code for a smooth type range block is very simple as every pixel value is replaced by the average of all the pixel values. So elements (pixels) which are included in the smooth type range block will converge right after first iteration. On the other hand, the edges, which are of main interest here, are included in the rough type range blocks and a GA based fractal encoding scheme is adopted for this type of blocks (see Chapter 2). In the process of encoding, rough type range blocks will match with domain blocks containing edges. Moreover matching of an edge pixel is restricted to

only edge pixels as we are interested in finding the similarities present in the image. In a rough type range block, in general, there will be both edge pixels and non edge pixels. We want to approximate all the pixels through a contractive continuous map. The closeness of the estimator with a pixel value will depend not only on the map but also on the complexity of the pixel (*i.e.*, the pixel value in relation to its surrounding pixel values) from where it is being mapped. As one edge pixel is mapped to another edge pixel (since we search for similarities), and, one map is not sufficient to carry the information regarding an edge pixel (since a domain block has edge and non edge pixels), it is expected that an edge pixel is approximated by a sequence of maps. Hence, the sequence of the form (4.2) for an edge pixel will have long length compared to that of a non edge pixel. Thus it is evident that edge pixels will require more iterations to converge compared to non edge pixels. Hence, on the basis of convergence time, we can classify the pixels as edge and non edge.

In support of our claim, we illustrate another phenomenon which occurs during the process of decoding. The decoding process of "Lena" image from its PIFS code has been stopped after ten iterations as no visual difference has been found in successive iterations. But from the difference images, as shown in Figure 4.5, it is observed that strong edges are prevalent, though with low intensity, even after ten iterations. These difference images are obtained by taking difference between original image and images obtained during decoding process. This phenomenon experimentally supports our claim that edge pixels take more time to converge than non edge pixels. Also this illustration indicates the edge information of an image can be extracted from its PIFS code from the reconstruction sequence.

Now the question remaining is: how to extract the edges from PIFS code. To solve this problem, we have introduced a penalty function on the pixels during the process of decoding (convergence) of PIFS code. The reconstruction sequence is modified to include a process of edge extraction. Reconstruction process has now two parts. One is decoding of PIFS code to get an image which is very close to the original one, the other one is the extraction of edges which results in an edge image. The process of edge extraction and the penalty function are discussed below.

We have already mentioned that edge pixels take long time to converge to their respective fixed points. So, to locate edge pixels, we have attached penalties to pixels, depending on their convergence status. The more the number of iterations to converge,

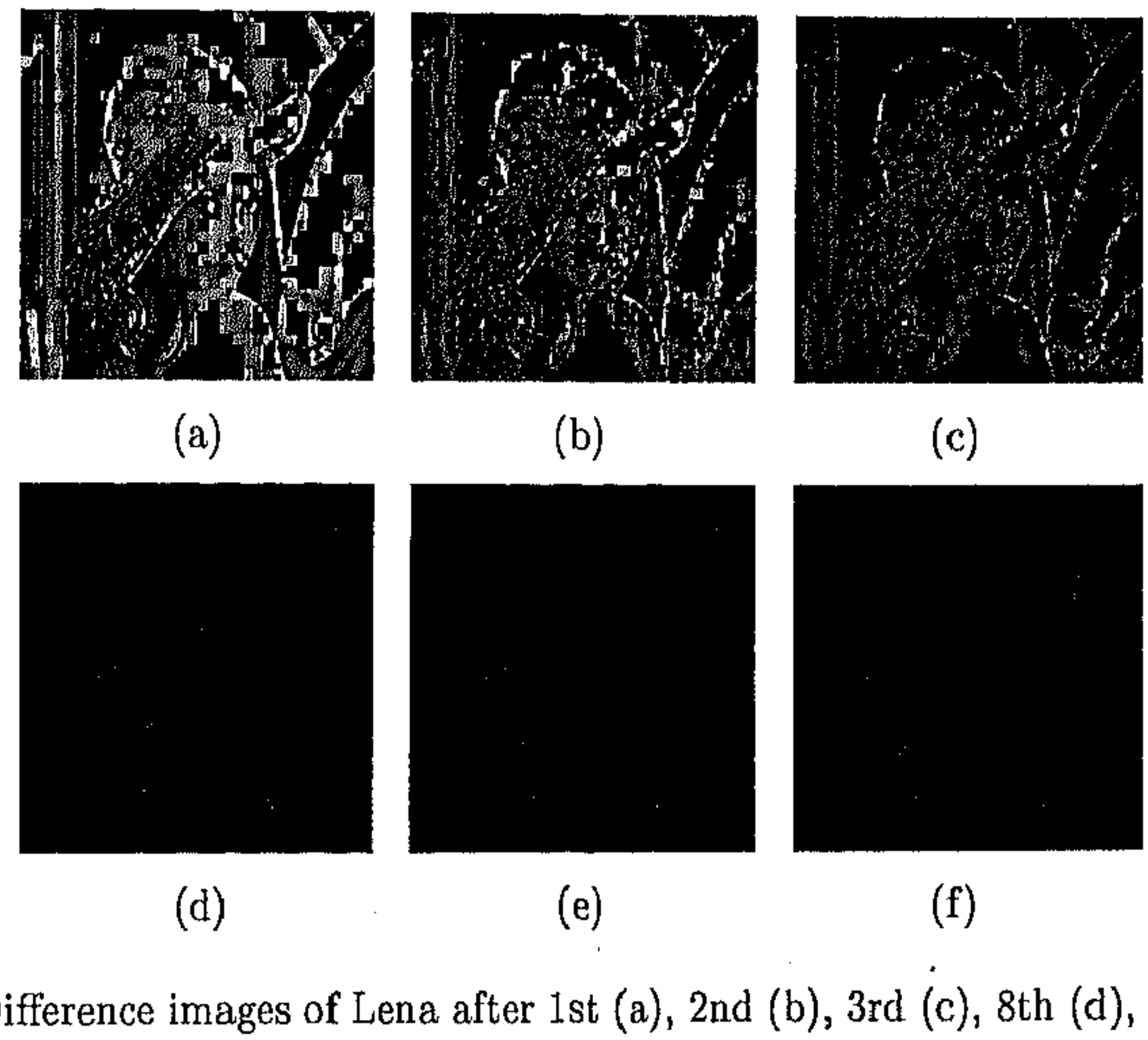


Fig. 4.5: Difference images of Lena after 1st (a), 2nd (b), 3rd (c), 8th (d), 9th (e) and 10th (f) iteration with starting image Rose

the more is the penalty attached to that pixel. Note that, PIFS code is nothing but a set of contractive maps which are continuous. But, edges present in the image are discontinuities due to sudden changes in gray values. During the process of decoding, PIFS code needs to grab those discontinuities by continuous maps. The process of approximating discontinuities by continuous maps is necessarily slow, *i.e.*, it takes more iterations to give rise to a better approximation. Hence, pixels having large penalty values would be edges pixels. The only limitation of this methodology is that the convergence of pixels depends on the initial starting image. Hence the penalty values of the pixels are highly dependent on the starting image. If the initial image is the image itself then all the pixel will converge in the first iteration during decoding. Even if the initial starting image is very close to the original one then also the penalty values of the edge pixels will not be too high to indicate the presence of edges. The present algorithm function well only if one starts decoding fractal codes either from a blank image or from an image which is mostly dissimilar compare to the original one e.g. in the present case, one may use "Rose" image as a starting image to get the edge image of "Lena".

We have designed here a very simple penalty function scheme using the number of iterations, which depends on the status of the convergence. Once the pixel value converges to a fixed point, no further penalty is imposed. On the other hand penalty of each iteration is added with the penalty that already exists for a pixel if the pixel value has not converged yet. The penalty values for the first few iterations need to be small to let the non edge pixels to converge. Similarly, the penalty values for the last few iterations need to be small so that the impact of the strong edge pixels on the other edge pixels would be less. These penalties for the pixels, once the number of iterations for reconstruction is fixed, would indicate the degree to which a pixel may be designated to be an edge pixel. Thus, the edge image is nothing but an image where the gray value of a pixel is nothing but its penalty value. So we will fix two iteration values which will be considered as lower and upper bounds and within these bounds, penalty values will be high. To represent this phenomenon, an S type penalty function scheme is used. The mathematical formulation of the penalty function is as follows.

Let $p(i,t)$ be the penalty function for a pixel at iteration i. Here t is difference of pixel values between ith and $(i-1)th$ iteration. $p(i,t)$ is defined as follows.

$$\begin{aligned} p(i,t) &= 0 && ;\ i=0\ ,\ \forall t \\ p(i,t) &= p(i-1,t) && ;\ i \geq 1\ ,\ t < \delta \\ &= p(i-1,t) + \sum_{j=1}^{i} j * c(j) && ;\ 1 \leq i \leq N\ ,\ t \geq \delta \end{aligned}$$

where the coefficient $c(j) \in [0,1]$ is computed from a Π type function [133] which is as follows

$$\begin{aligned} c(j) &= 1 - 2 * d(j)^2 && ;\ 0 \leq d(j) < \frac{1}{2} \\ &= 2\,(1-d(j))^2 && ;\ \frac{1}{2} \leq d(j) < 1 \\ &= 0 && ;\ Otherwise \end{aligned}$$

where

$$d(j) = \frac{|j - \frac{L_1+L_2}{2}|}{L_2 - L_1}.$$

Here,

δ = Threshold value of t beyond which penalty is non zero

L_1 = Lower bound for the number of iterations

L_2 = Upper bound for the number of iterations

N = Number of iterations at which sequential process is stopped

These parameters δ, L_1 and L_2 act as control parameters of the proposed scheme. Users can get their desired output by adjusting these parameters. For example, if the compression ratio is low one has to fix a large value for L_1, otherwise it may lead to an image with many undesired edges. So, one can play with these parameters and may explore some other interesting results.

Incorporating the penalty function in the reconstruction process, an edge image of the original image can be obtained. The entire process can be viewed as a block diagram given in Figure 4.6.

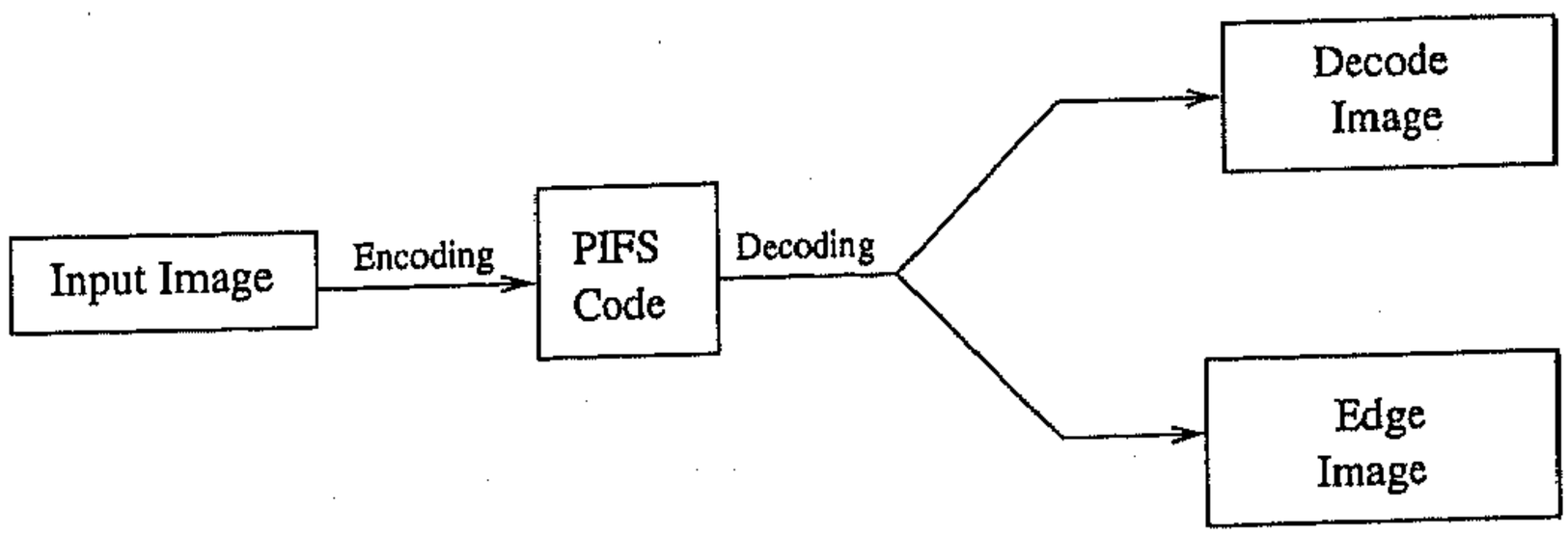


Fig. 4.6: Block diagram of the reconstruction process of PIFS code resulting in either decoded image or edge image

Now, it is observed that, in many cases, the output edge image which is obtained from reconstruction sequence with penalty function may have discontinuity in edges. So, we have proposed a very simple edge linking algorithm to get the final edge image.

4.4.3 Edge Linking

In the above mentioned edge extraction algorithm, the edge image comprises of penalty values associated with each pixel. In this process, some discontinuities may appear at different edge segments. To overcome this we have introduced an edge linking procedure. The proposed edge linking procedure is not a general edge linking mechanism. It is applicable for this type of special cases only. This edge linking procedure is based on a convolution operation with a set of predetermined kernels (windows) of the same size but having different coefficients. The penalty value of the pixel which is at the center of kernel is modified in course of sliding the kernels over the entire image.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Fig. 4.7: 3×3 kernel with coefficients

In particular, we have considered a set of four kernels to find out responses in four directions, namely, horizontal, vertical and two diagonals. The output obtained from the convolution process is called response of a kernel. The maximum response is determined and the center pixel is modified based on this maximum response. For a 3×3 window, as shown in figure 4.7, the main task involves computing the sum of products of the coefficients (w_i) with the penalty values (p_i) contained in the region encompassed by the window. Thus the response of a kernel for a point at its center is given by

$$w_1p_1 + w_2p_2 + \cdots + w_9p_9 = \sum_{i=1} w_ip_i \cdot$$

Let the response of the kth kernel for its center pixel be denoted by $E_k; k = 1, 2, 3, 4$. The response of a kernel, centered at a boundary pixel, is computed by using the appropriate partial neighborhood. The values of coefficients (w_i) are different for different kernels. For four kernels the values of coefficients in four directions are as follows.

$$\text{Horizontal} \;:\; w_4 = w_5 = w_6 = 1, w_1 = w_2 = w_3 = w_7 = w_8 = w_9 = 0$$

$$\text{Vertical} \;:\; w_2 = w_5 = w_8 = 1, w_1 = w_3 = w_4 = w_6 = w_7 = w_9 = 0$$

$$\text{First Diagonal} \;:\; w_3 = w_5 = w_7 = 1, w_1 = w_2 = w_4 = w_6 = w_8 = w_9 = 0$$

$$\text{Second Diagonal} \;:\; w_1 = w_5 = w_9 = 1, w_2 = w_3 = w_4 = w_6 = w_7 = w_8 = 0$$

The choice of the values of the coefficients are similar to that of line detection in four directions [65]. Here p_5 is the penalty value corresponding to the center location of the kernel. Let $E_{max} = MAX_i\{E_i\}$. Compute $A = \frac{E_{max}}{r}$, where, r is the number of non zero coefficients of the kernel. In the present case $r = 3$ for all kernels. Now the central penalty value (p_5) is modified according to the following rule.

$$\begin{aligned} p_5 &= p_5 \;;\; \text{if } p_5 > A \\ &= A \;;\; \text{if } p_5 \leq A \,. \end{aligned}$$

The final output edge image is thus obtained from PIFS code by applying a decoding scheme with penalty function followed by the edge linking algorithm. The block diagram shown in Figure 4.8 provides the overview of this whole process.

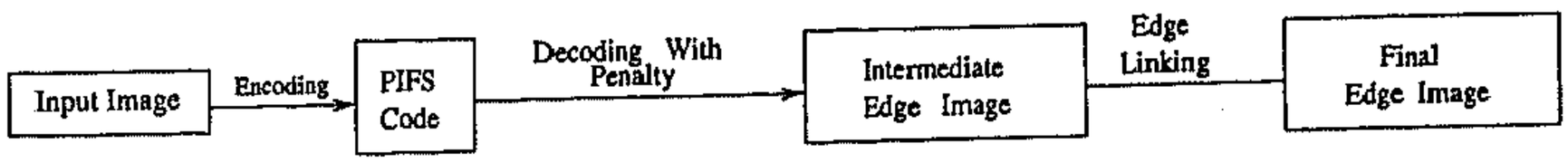


Fig. 4.8: Block diagram of reconstruction process of PIFS code resulting in edge image

The results of the specific implementation of the aforesaid algorithm are given in the next section.

4.5 Implementation and Results

To find the PIFS or fractal code for a given image, the search is to be made for all possible domain blocks as well as eight possible isometric transformations [80]. To

reduce the search space and time Genetic Algorithm is used instead of exhaustive search [see Chapter 2].

For the specific implementation of the proposed algorithm, two types of images are considered. A synthetic image ,"Circle" image as shown in Figure 4.9, and some real life images are treated as original input images. In particular, real life images which are used as input to the GA based fractal image compression scheme described in Chapter 2 are used here also. These original figures are given in Chapter 2.

The "Circle" image is a 128×128, 8 bit/pixel image and on the other hand real life images are all 256×256, 8 bit/pixel images. Here we have considered a synthetic "Circle" image to judge the performance of the proposed edge extraction algorithm. As there are hardly any measures to judge true edges in a scene, we have tested our algorithm on a synthetic image where the true edges are easily distinguished from non edges by human eyes. The PIFS codes of both the images are obtained by the GA based technique mentioned in Chapter 2. We have considered range blocks of size 8×8 and 4×4 (two level) for the computation of PIFS code. The results of decoded images of "Circle" is shown in Figure 4.10. Other decoded images have already been shown in Chapter 2.

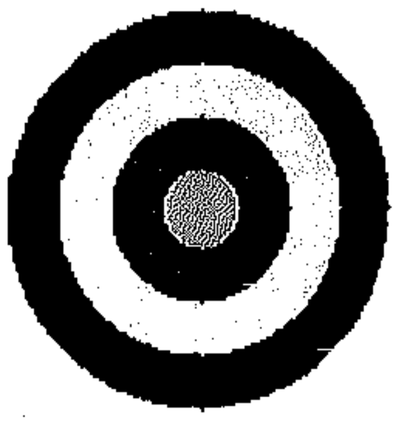


Fig. 4.9: Original Circle image

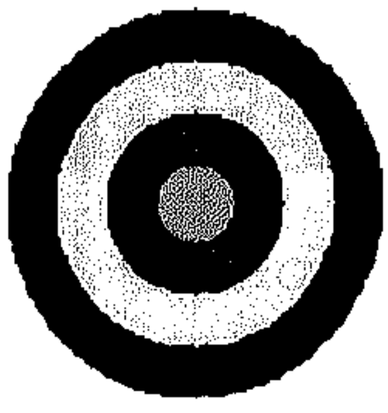


Fig. 4.10: Decoded Circle image

The obtained PIFS code of "Circle" image is now decoded with the proposed penalty function scheme followed by proposed edge linking procedure with window of size 5×5. Other parameters of edge linking algorithm are set experimentally to get the best results. It has been found experimentally that after 20 iterations most of the image becomes stable. So, we set $N = 20$. Values of other parameters are $T = 0.1$, $L_1 = 3$ and $L_2 = 17$. Edge images of "Circle" with edge linking is shown in Figure 4.11. PIFS codes of real life images ("Lena", "LFA", "Seagull" and "Girl") are also processed similarly with a window of size 11×11 for edge linking. Note that the size of the

window is set heuristically. The values of the other parameters are same as before. The output edge images of "Lena" without edge linking and with edge linking are shown in Figures 4.13and 4.14 respectively. The resultant edge images with edge linking of "LFA", "Seagull" and "Girl" are shown in Figures 4.16, 4.18 and 4.20 respectively.

In practice, the results of edge extraction methods are presented as binary images. Usually, an edge pixel is represented by "1" and non edge pixel is by "0". This process is known as thresholding. To represent the results of the proposed technique in thresholded form, we have set a threshold value "30" for "Circle" image and the threshold value of "100" for the real images. The values are set heuristically to obtain the best results. But there are several techniques of thresholding and the users of the proposed technique may choose the methodology of their own choice. The obtained results of thresholded edge image of "Circle" is shown in Figure 4.12. The same for "Lena", "LFA", "Seagull' and "Girl" are shown in Figures 4.15, 4.17, 4.19, and 4.21 respectively.

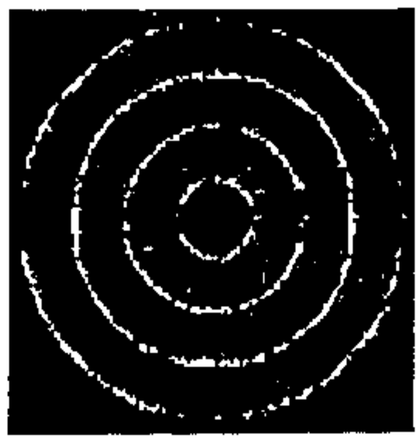


Fig. 4.11: Edge image (with edge linking) of Circle

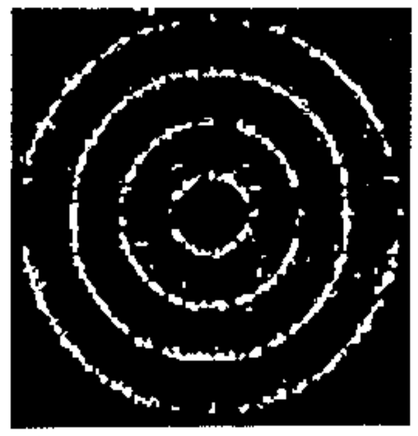


Fig. 4.12: Thresholded edge image of Circle

It may be noted that in all conventional edge detection schemes, the whole image is convolved with kernels of different sizes and coefficients based on the application demand. These schemes, generally, try to find out the gray value differentials at different portions of the image. So, an edge detector is nothing but a tool to find out gray level variations. But in the proposed technique there is no such kernel, as in the case of conventional edge detector, for edge extraction. Rather edges are extracted, indirectly, during fractal reconstruction from auxiliary information like rate of convergence and mode of convergence of PIFS code. The proposed scheme performs edge extraction task in the compressed domain. The compressed domain image processing has its own limitation of loss of information during the process of encoding. A trade-off in

Fig. 4.13: Edge image (without edge linking) of Lena

Fig. 4.14: Edge image (with edge linking) of Lena

Fig. 4.15: Thresholded edge image of Lena

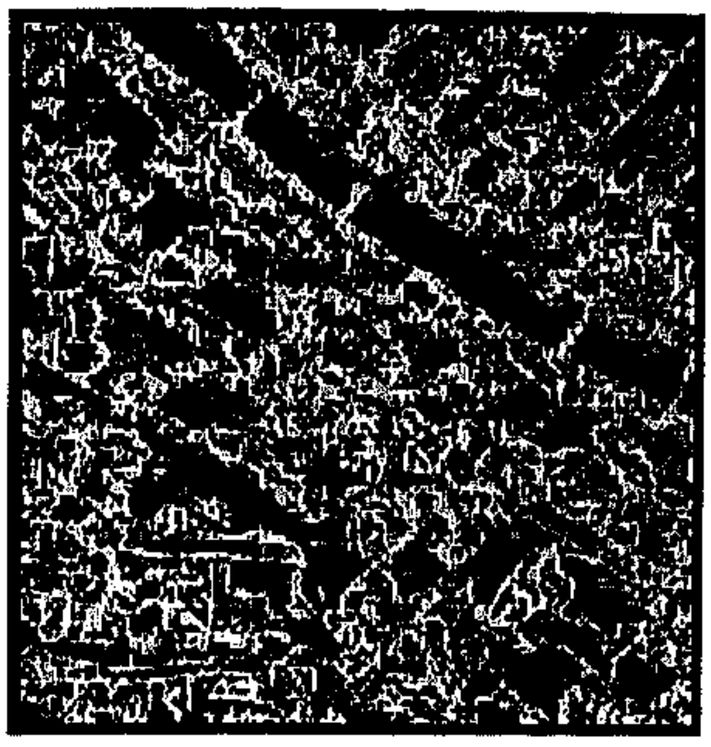


Fig. 4.16: Edge image (with edge linking) of LFA

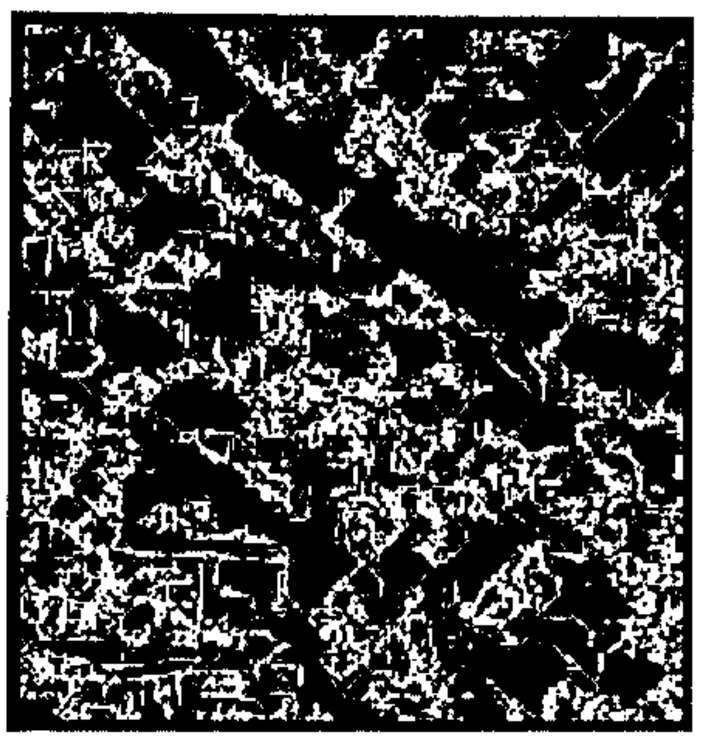


Fig. 4.17: Thresholded edge image of LFA

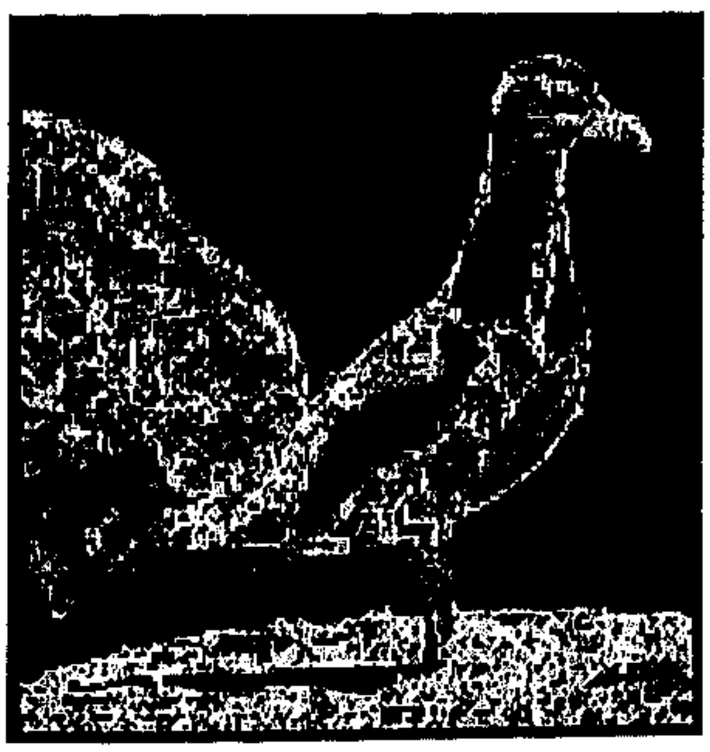


Fig. 4.18: Edge image (with edge linking) of Seagull

Fig. 4.19: Thresholded edge image of Seagull

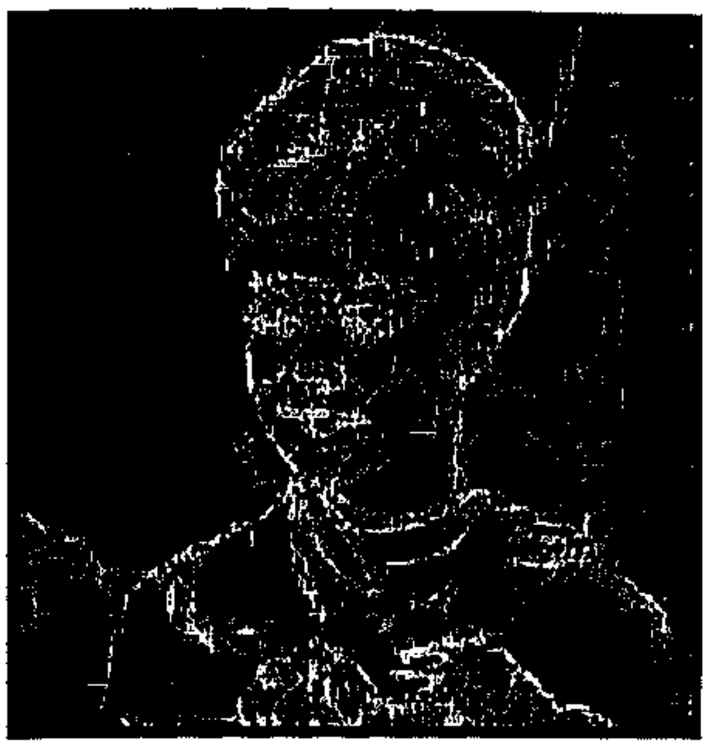


Fig. 4.20: Edge image (with edge linking) of Girl

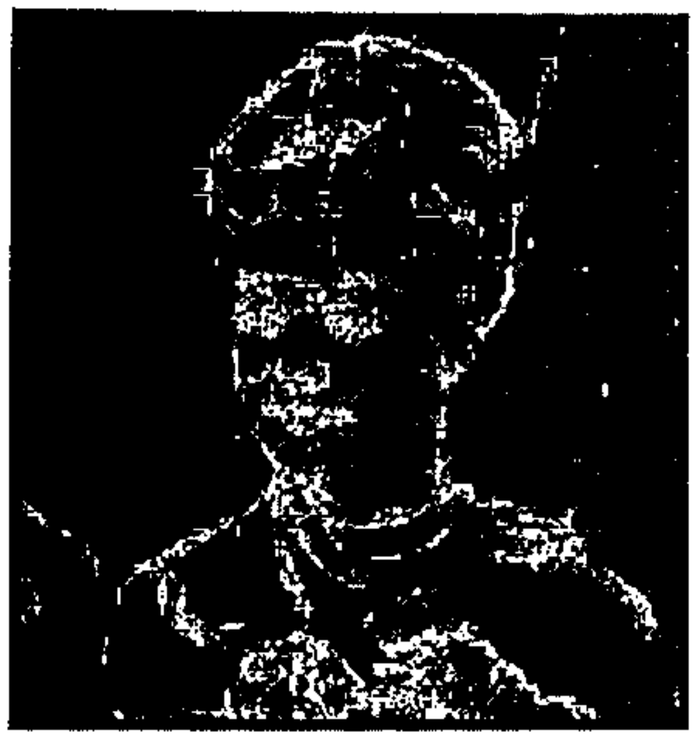


Fig. 4.21: Thresholded edge image of Girl

between advantages of the algorithm and loss in quality would be appreciated during compressed domain image processing. Our aim, in the present chapter, was to investigate the applicability of the fractal code for performing image edge extraction. At the present moment, we are not benchmarking the proposed edge extraction scheme with the conventional one from the stand point of conventional parameters like noise sensitivity, rotational sensitivity and so on.

Now for visual comparison we are showing here the results of a standard edge detector which is "Canny" edge detector [28]. Among the techniques available in the literature on edge detection, Canny operator is considered to be the most robust one. The Canny operator is applied on the "Original Lena" image and on the "Decoded Lena" image which is obtained from a GA based PIFS code of "Lena". Figure 4.22 and 4.23 are respectively the results of Canny operator on "Original Lena" and "Decoded Lena".

4.6 Conclusions and Discussion

The most important advantage of the proposed technique of image edge extraction using PIFS code is that it utilizes the coded (fractal) version of the input image instead of the original image. Therefore it is cost effective in the sense of storage space and time as no decoding is performed before using the edge extraction algorithm. Moreover, the

Fig. 4.22: Result of Canny on Original Lena

Fig. 4.23: Result of Canny on decoded Lena

proposed scheme can directly be used on the images accessed from the digital library where usually images are kept in compressed form. To implement the present algorithm, the only restriction one should impose is that the initial starting image should be either a blank image or an image which is mostly dissimilar compared to the original.

The process of edge extraction is also valid in case of condensation maps included in the PIFS code. It has been discussed in Chapter 2 that there are two types of condensation maps and these are contractive as well. A range block is reproduced as it is using the corresponding condensation map of the first kind. On the other hand, each pixel value of a range block, corresponding to a condensation map of the second kind, is replaced by the average of pixel values of that range block. The first type of condensation map can be used for any range block where as a condensation map of the second type is defined only for the smooth type range blocks [see Chapter 2]. During reconstruction, pixel values of a range block will converge to their fixed points, immediately after the first iteration, under a condensation map. Thus, here, only the second type of condensation maps could be used for the extraction of edges. Note that, in the present thesis, to generate PIFS codes as described in Chapter 2, only the second type of condensation map is used.

It has been found that there are some differences, visually, in the edges extracted by the proposed scheme on the compressed images with the Canny operator on the original images. The edges extracted by the proposed scheme are smudgy where as the edges

found by the Canny operator are smooth. Otherwise it may be observed that, the edges obtained by the Canny operator are more or less present in the edge images of the proposed scheme.

The proposed edge extraction method is not the panacea of all the problems of edge detection. The results, which are promising, are being reported here. There is a lot of scope for improvement. The design of penalty function, applied in the reconstruction sequence, is very simple. A more generalized penalty function, representing an edge pixel in a better way, needs to be found. Such a function may make the task of edge linking redundant.

The proposed scheme is highly dependent on the compression scheme adopted. There are several compression schemes available in the literature. But the present edge extraction scheme is applicable only on the fractal code. The proposed edge extraction scheme is not applicable with other compression schemes like run length coding, vector quantization coding and wavelet based procedures.

The proposed technique is noise sensitive while most of the edge detection algorithms are noise insensitive. But this problem can not be avoided as we are using coded version of the image instead of the original one. If noise is incorporated in the original image then the coded version will usually inherit that noise and hence it will be reflected in the edge image obtained by this algorithm. This problem can be avoided in two ways. One can adopt an encoding mechanism which is noise insensitive or apply a noise removal algorithm before doing the encoding. The problem of noise sensitivity is a general problem in the field of image processing in compressed domain.

The performance of proposed edge extraction algorithm is highly dependent on the PIFS or fractal code. It implies that if a fractal code reproduces a good quality reconstructed image, then only one can expect a good quality edge image. So, utmost care should be taken in the encoding process. From figure 4.22 and figure 4.23, it has been found that the results of Canny operator on original and decoded "Lena" are almost same. Thus, it indicates the high performance of the GA based fractal image compression technique.

Chapter 5

Conclusions and Scope for Further Research

The thesis deals with fractal based representation of images to perform various image processing tasks like image compression, image magnification and image edge extraction. Representation of an image in terms of a set of affine contractive maps has been regarded as the fractal representation. The set of transformations is called iterated function system (IFS). However, the major contribution of the thesis is based on partitioned iterative function system (PIFS). The basic difference between IFS and PIFS exists in the domain of application of their respective transformations. The work carried out in the present thesis is summarized briefly below, followed by conclusions and discussion.

5.1 Summary of the Work Done

- In Chapter 2, a Genetic Algorithm based PIFS technique for image compression is proposed. The method is found to be computationally efficient in searching appropriate transformations for small non-overlapping image blocks. The GA based fractal image compression method has been implemented on colour images too. This chapter also deals with the fractal based methodology to encode one dimensional signals. In particular, the image compression technique has been modified suitably to encode EEG signals.

- A methodology for performing image magnification task which uses coded form of the image has been developed in Chapter 3. The code used here is obtained by the PIFS based technique as described in Chapter 2. No magnification operator like interpolation is needed here. The performance of the said methodology has been judged by two newly developed distortion measures.

- A new image edge extraction algorithm has been developed in Chapter 4. The technique is embedded in the process of reconstruction of an image from its fractal code. After reconstruction, instead of getting a decoded image, an edge image of the original is obtained.

5.2 Conclusions and Discussion

The prime contribution of the thesis revolves around the PIFS (fractal) codes of images. The PIFS code of an image is generated using a Genetic Algorithm based technique. The performance of the GA based technique in the context of image compression is found to be quite satisfactory. The overall reduction in the search space is found to be of the order of 20 times. The compression ratios and the quality of the decoded images are found to be comparable with that of the other existing fractal based methods. In many cases better performance has also been recorded. Note that to implement GA based scheme we have used non overlapping square partitioning of range blocks though GA based searching, described here, can also be utilized in case of other range block partitioning *e.g.*, quadtree, HV, triangular and irregular.

The effectiveness of the GA based fractal image compression technique depends on mainly two factors. 1) The amount of self similarity present in the given image and 2) the proper selection of the parameter values in GAs.

Here, self similarity implies the fact that under suitable affine transformations, the pixel values of larger blocks (domain blocks) of the image match with the pixel values of smaller blocks (range blocks) in the same image. Thus, if there exists a perfectly matched domain block and a suitable affine contractive map for each range block, the performance of the PIFS based technique would be at its best. Now, if a suitable affine transformation for a particular range block is not available using any domain block, then the quality of the decoded image would be poor. To get suitable transformations or to get a good quality decoded image, a reduction in the size of the range block would be helpful. Note that, if the range block size is 1×1, then for every domain block of size 2×2, there exists an affine contractive transformation such that the approximation of the range block is error free. But, there will be no compression if all the range blocks have size 1×1. Thus, for the selection of the size of the range block, a trade off has to be made between obtaining a good quality decoded image and a high compression ratio.

A condensation map of the range block, whenever necessary, could also be considered if the approximation of the range block is of poor quality. Condensation maps take care of the image regions which are smooth. The pixel values are almost same in a smooth type range block. A simple classification scheme has been utilized to search

out smooth image blocks. Finally, it can be concluded that although the intuition behind PIFS scheme is based on self similarity, it is not actually needed for applying the present algorithm.

The other factor on which the performance of the GA based PIFS technique hinges upon is the proper selection of the parameter values in GA *viz.*, (i) the size of the initial population, (ii) the number of iterations performed, iii) crossover probability and iv) mutation probability. The selection of the values of these parameters is not unique. In fact there is no firm rule in GAs to select the parameter values. The values are usually selected on the basis of the demand of the problem concerned. However, for any choices of the initial population, crossover probability and mutation probability, the convergence of the GAs to the optimal solution is confirmed if the number of iterations is sufficiently large [23]. Note that, the number of points to be searched will be different for images having different sizes. The values of the parameters in GA thus need to be selected accordingly to get a good solution for a particular image. Our GA based method can provide efficient and fast encoding by adjusting the initial population and the number of iterations in case of large data sets.

Another important factor that influences the performance of the proposed technique is the choice of fitness function in the GA. This function is responsible for proper approximation of the range blocks. The overall effect of it can be visualized in the quality of the decoded image. Root mean square error (RMSE) is used here as the fitness function. Any other suitable function, which measures the distortion between the given range block and the obtained range block, can be used as the fitness function instead of RMSE in the proposed method. Moreover peak-signal-to-noise ration (PSNR) which is a function of RMSE is used here to measure the quality of the decoded image. There are other measures too in order to judge the quality of the decoded image [39]. It is easy to modify the proposed method when some other measures are used for judging the quality of the decoded image.

The said GA based method has been modified suitably to encode one dimensional signals. So far, very few attempts have been made to utilize fractal techniques for signal compression. The fractal based methodology described is found to be efficient, both in terms of compression and quality, for encoding EEG signals. The compression ratios are found to be better in comparison with those in [70, 171]. The quality of the decoded signals has been assessed both quantitatively and clinically. It has been

observed that the same clinical conclusions can be drawn using the decoded signals instead of original.

The performance of the EEG signal coding scheme has been examined by varying the sizes of the range segments. Compression ratio can be increased by selecting proper range segment size. By proper range segment size we mean that particular size of the range segment for which the finer details of the EEG signal are preserved for clinical diagnosis. Compression ratio is likely to be raised if the number of transformations is reduced. In this regard, it has to be assured that the fidelity of the reconstructed signals should not be affected. One may use fewer number of transformations instead of eight transformations for this purpose. In such a case, the selection of transformations will play a major role.

The thesis also dealt with some other image processing tasks such as image magnification and image edge extraction using PIFS code. These investigations regarding the applicability of the fractal technique, in particular PIFS based technique, to other possible areas of image processing are unique of their kind. Very few contributions have been made in this area. The most important feature of the proposed techniques of fractal image magnification and fractal image edge extraction is that these utilize the coded (fractal) version of the input image instead of the original one.

The algorithms developed in Chapter 3 and Chapter 4 using fractal code can also be viewed as compressed domain processing. These methodologies are highly dependent on the compression procedure adopted. There are several compression methods available in the literature. The proposed magnification and edge extraction techniques are applicable only on the PIFS codes. Any other code obtained by other compression schemes, like run length coding, vector quantization coding or wavelet based image coding, can not be used as input for the proposed algorithms.

The sizes of the range blocks considered play a vital role in image compression, magnification and edge extraction using fractal technique. The last two algorithms are in opposite direction of the first one from the point of view of range block size. In particular, finer image detail can be retained by using range blocks of smaller size and more compression can be achieved through range blocks of larger size. While performing the magnification and the edge extraction task it is very important to use the information regarding finer image detail. So, one can think of an optimal range block size for which

good quality magnified images and edge images can be reconstructed from the fractal codes and at the same time considerable amount of compression can be achieved.

Although the fractal image compression algorithm developed in Chapter 2 is based on block coding, the blocking effect is reduced to a great extent, as it tries to approximate image regions. In this connection, it should be noted that in fractal image magnification, images are magnified by expanding their fractal codes which are independent of image resolution. Our magnification method provides an approximation of the image blocks resulting in a sharper expanded image. Other image magnification techniques using pixel replication make magnified images blocky, blurry and patchy after a certain extent of expansion.

On the other hand, it has been found visually that there are some differences in the edges extracted by the proposed edge extraction method (Chapter 4) with that obtained by the Canny operator on the original images. The edges extracted by our method are smudgy whereas those found by the Canny operator are smooth. However, it is observed that, the edges obtained by the Canny operator are more or less present in the edge images of the proposed algorithm.

Since the coded version of the image is used as input, the said edge extraction technique is noise sensitive. If the input image is noisy, the coded form will usually inherit that noise and hence it will be reflected in the edge image. This problem can be avoided in two ways. One can adopt an encoding mechanism which is noise insensitive, or apply a noise removal algorithm before encoding.

5.3 Scope for the Further Research

The most important problem, dealt within the thesis, is the generation of the PIFS code. Only the affine contractive maps are considered for computing PIFS codes. Moreover, out of all possible isometries, only eight have been selected for our investigation. A study of PIFS using other affine transformations would therefore be appreciated. The performance of the fractal coder using nonlinear transformations also needs to be examined.

Genetic Algorithms have been used in the present investigation in order to reduce the search space for finding suitable transformations for non-overlapping square im-

age blocks. The performance of GAs can be examined for other partitioning schemes. For example, quadtree can be applied during the partitioning of the given image and GAs can be applied as the search mechanism in each step of quadtree partitioning. GAs can also be used in other partitioning such as horizontal-vertical (HV), triangular and mixed partitioning. One may also probably use other search techniques like simulated annealing, tabu search, apart from GAs, and their hybridization methods, for conducting the search in a short time.

In our GA based fractal compression method (Chapter 2), a simple classification mechanism of the range blocks is introduced. Some modifications can be incorporated in it. The threshold value for the classification corresponds to the valley in the histogram. However, it is not always assured that the said histogram is strictly bimodal. The suitable values for the valleys in the histograms are obtained here by visual inspection. The problem of thresholding a histogram is very old and there are innumerable articles on this from different points of view [128, 132]. One can use any one of the several thresholding methodologies for finding valleys in the histograms.

In the encoding process of one dimensional signals, only normal EEG during sleep and wakefulness has been tested. This methodology may be applied to the EEG in other states, like, epilepsy and depression. Moreover, evoked EEG responses too, may be compressed using this method.

Usually, EEG signals are very irregular in nature and research is going on to find important features to identify the signals [113, 153]. A study towards the selection of important features of EEG using PIFS code would thus be appreciated.

The work that remains to be performed for the magnification of an image, as described in Chapter 3, is the magnification by any order including fractional orders. A possible guideline towards the solution of this problem has been discussed. But, this may suffer due to discretization. A detailed study on the magnification by any order requires further research.

The proposed edge extraction method (Chapter 4) is not the panacea of all the problems of edge detection. The results are very promising and have been reported here. There is a lot of scope for improvement. The design of penalty function, applied in the reconstruction sequence, is very simple. A more generalized penalty function needs to be found in such a way that it produces smooth edges and, in the process, making the

task of edge linking redundant.

The study of the convergence of the PIFS coder at pixel level in the context of image edge extraction (Chapter 4) resulted in an interesting observation. It has been observed that, under PIFS framework, pixels of an image are related through affine contractive transformations in a chain like sequence. Although this inter pixel relation is exploited to develop an edge extraction algorithm (Chapter 4), still there is a lot of scope for synthesizing this relation for further image analysis.

An investigation on the effectiveness of the PIFS based technique for performing image magnification and edge extraction has been made in this thesis. Other tasks like image enhancement and segmentation may also be tried in the said fractal framework.

Appendix A

Basic Principles of Image Compression, Magnification and Edge Extraction

The problem of image compression, magnification and edge extraction are described here. The methodologies developed for solving these problems have not been described below. A brief glimpse of the problem of finding a few solutions to these tasks has been provided.

A.1 Image Compression

The sheer volume of digital images, represented as a two dimensional function of pixel values, can be alarming due to its memory requirement. In fact, the amount of data generated from digital images may be so huge that it may give rise to impractical storage, processing and communication requirements. In such cases, representation of images other than the usual pixel form is needed. Now onwards, the usual pixel form of the image will be referred to as image data. The basic feature of an image compression problem is the removal of redundant information present in the image. There are three basic redundancies, *viz*, inter pixel redundancy, psychovisual redundancy and coding redundancy which are exploited in image compression problems. Along with the removal of redundant information (encoding), an image compression technique consists of retrieval of the original image information or an approximation of it (decoding). Image coding and decoding are the two complementary operations of an image processing algorithm. The encoding operation is performed prior to storage or transmission of the image information. On the other hand decoding of the compressed image information is carried out prior to performing any image analysis task or at the receiving end in case of transmission. The role of image compression is unquestionable in many important and diverse applications including multimedia, remote sensing, document processing and medical imaging, televideo conferencing, facsimile transmission (FAX) and so on. The list in fact is endless, and the ever expanding number of applications need efficient image compression techniques. The efficiency is decided by the storage of compressed information with low cost and regeneration of images with acceptable or high quality.

In any compression technique the most important task is to design an encoder and a decoder (an inverse of the encoder). The activity of the encoder can be divided into three components which are as follows.

- *Transformation :* In this stage the input image data is transformed into a new

form capable of reducing interpixel redundancies in the input image. This operation is reversible and may sometimes lead to straight forward reduction in the memory requirements. Run-length-coding is an example such case. Discrete cosine transform (DCT), wavelet transform and fractal transform are few examples of transformation operations. Here the image is represented by a set of coefficients evolved in the process of transformation of image data.

- *Quantization :* In this stage, the output of the first stage (coefficients of the transformation) is converted into suitable discrete values. Usually, the coefficients are real numbers. These real numbers are quantized and represented by discrete numbers usually called symbols. The psychovisual redundancies are reduced as much as possible in this process. But the quantization is an irreversible process. The loss incurred during this process is optimized according to some prespecified constraints regarding the quality of the reconstructed image.

- *Coding :* The last and final stage is the allocation of symbols in bits which are the most fundamental form of storing data in computers. The coding redundancies of symbols are reduced in this stage.

The processes of transformation, quantization and coding constitute the encoder of a compression technique. But all three processes are not necessarily included in all the compression techniques, *e.g.*, the process of quantization is not included in the encoder if a lossless compression is required. The process of decoder consists of inverse operations of the encoder process and these operations are performed in reverse order. The inverse operations of coding, quantization and transformation are performed one by one. Note that in each step of encoding complete information of the process should be kept to perform the inverse operations during decoding.

Compression techniques are broadly classified into two classes. One is the lossless compression and the other is the lossy compression. Detailed descriptions of existing image compression algorithms are available in [37, 82, 145, 157, 65]. In lossless compression, the entire information of the original image data is retrieved from the compressed data. Hence it is usually used for really high quality or critical applications such as archival of medical and business documents. As it has already been mentioned, a lossless compression technique is composed of two operations, one is the representation of image

data into a suitable form using some transformations and the other is the coding of the transformed representation. Some commonly used methods of this kind are

- Run-length-coding
- Arithmetic coding
- Huffman coding
- differential coding
- contour coding

In lossy compression, only the perceptually relevant and significant information of images are regenerated form the compressed representations. Unlike the lossless technique, a lossy technique usually achieves more compression at the cost of the quality of the reconstructed image. In most of the cases, the resulting distortion of original image with its approximation, which is regenerated form the compressed information, is very low in respect to the visual judgments. All the existing compression techniques of this kind may be classified as

- Predictive coding
- Transform coding

The basic difference between lossless and lossy compressions is the presence or absence of the quantization stage of the encoding process. There are other lossy compression techniques where the first stage, *i.e.*, the transformation part, is missing. Vector Quantization (VQ) method of compression [53, 63, 102, 166] is such a coding technique. Note that the fractal compression technique, which is one of the major aspects of the present thesis, is included in the lossy transform coding.

Recently, another image compression mechanism based on wavelet transform has become very popular. Some basic concepts of wavelets theory are available in [35]. The wavelet transform provides an efficient representation of images causing reduction in storage requirement [1, 5, 104]. Most of the wavelet coefficients of an image would be

nearly zero, and the image thus would be well approximated with a small number of large wavelet coefficients.

In the literature on image compression, a different kind of coding mechanism is also available. Two different coding schemes are appended to result in a new coding scheme capable of performing better in the sense of compression ratio, quality of the decoded image and computational cost. This special type of coding scheme is known as hybrid coding [66]. A review of image data compression can be found in [94].

Another important issue related to image compression is the image coding standard. Regarding the performance of image compression techniques, some standards have been developed by various international organizations. There exists different standards with different motivations. Among these, JPEG standard is used for still image compression. There are other standards for colour image coding, image sequence (movie) coding and so on. Details of these standards are available in [36, 138, 174]. Note that the performance of the fractal image compression techniques are well up to the standard.

A.2 Image Magnification

Image magnification, in connection to the digital images, is an important task performed with two different motivations . The first one is to increase the image resolution such that the implicit information present in the original can be explicitly visible. The second one is to bring image data, from different sources, to a common scale. The applications of image magnification include satellite image analysis, medical image display and matching of images captured using sensors having different capturing resolutions [32, 176]. The main motivation for developing image magnification algorithms is to emulate the ease and perfection of optical magnification techniques. Most of the digital magnification techniques are basically surface interpolation methodologies based on linear, bilinear, cubic or bicubic interpolation [65, 88, 136, 159]. But the problem of this kind can also be solved by approximation techniques.

The mathematical description of the problem of image magnification is as follows. Suppose a set of points $\{(x_i, y_i);\ i = 1, 2, \cdots, n\}$ is given where $(x_i, y_i) \in \mathbb{R}\ \forall i$. Let there exists a function f such that $f(x_i) = y_i$ for $i = 1, 2, \cdots, n$. Let the functional form of f be unknown. Let $x_i < x_2 < \cdots < x_n$. The problem is now to find the value of f at

$x = a$ where $x_1 < a < x_n$. An interpolation technique finds a functional relationship f^* between y and x such that $f^*(x_i) = y_i, i = 1, 2, \cdots, n$ and then determines the value at $x = a$ by computing $f^*(a)$. Note that f^* may not be identically equal to f. On the other hand approximation technique defines a functional relationship between y and x such that an error function defined on the given set of values and the estimated set of values is optimized. A digital image magnification problem can be viewed as an image reconstruction problem where some extra spatial coordinates, other than the given coordinates, are to be defined and the pixel values corresponding to new coordinates are to be determined using either interpolation or approximation techniques. The ratio of the size of magnified image to that of original image is known as magnification factor. Although, theoretically, the value of the magnification factor could be anything, practically it should ensure the size of the magnified images to be an integer so that the problem of discretization regarding the storage of magnified images can be avoided. Magnification techniques using interpolation and approximation have their limitations too. Most of the existing techniques of magnification suffer from artifacts due to blocking, blurring and ringing in case of large magnification factors.

A.3 Image Edge Extraction

Edge extraction of an image is a preprocessing task performed for detecting discontinuities present in the intensity values of the image for image segmentation purpose. Image segmentation refers to the area of processing known as image analysis. Segmentation subdivides an image into its constituent parts or objects. Segmentation algorithms of digital images generally are based on two basic properties of pixel values. These are discontinuity and similarity. Segmentation using discontinuity is based on abrupt changes in pixel values. Areas of interest of this kind of processing include extraction of isolated points, extraction of lines and edges of images.

An edge is the boundary between two regions having distinct pixel values. To define a region, it is assumed that the pixel values of the region are as homogeneous as possible. Based on this assumption the transition between two regions is determined on the basis of discontinuities in the pixel values. Hence, edges represent an important feature of the image as changes or discontinuities in image features like intensity are called edges. The process of extraction of such image features is called edge extraction. The process, in

the broad sense, can also be called as feature extraction method. In the process of edge extraction, an edge image, which indicates the presence or absence of edge points, is produced from a gray level image. A great deal of effort has been made towards finding the solution of this problem [28, 30, 68, 93, 110], but considerable success is yet to be achieved as edge semantics are extremely complicated. The understanding of proper edge is still inadequate as there is hardly any measure which emulates human visual system to judge the true edge in an image. The applications of edge extraction include various problems of computer vision. It can also be applied for pattern recognition and pattern matching problems.

The underlying idea of most of the edge extraction algorithms is the computation of a local derivative operator. Commonly used algorithms are based on finding local gradients typically computed over a small window which scans the entire image. Generally, the sizes and the coefficients of windows are different for different edge detectors. These windows are called gradient operators and the corresponding processes are known as edge detectors. The edges are computed from the gradient information, obtained by gradient operators, using thresholding based approaches.

Bibliography

[1] E. H. Adelson, E. Simoncelli, and R. Hingorani, "Orthogonal pyramid transform for image coding," in *Volume 845 of SPIE Proceedings*, (Cambridge, USA), pp. 50–58, 1987.

[2] M. Ali and T. G. Clarkson, "Survey of block based fractal image compression and its applications," in *Proceedings of the 2nd Seminar on Information Technology and its Applications (ITA'92)* (K. Hafeez, M. A. Wani, and K. S. Jomaa, eds.), pp. 110–122, Leicester, 1992.

[3] M. G. Alkhansari and T. S. Huang, "Fractal-based techniques for a generalised image coding method," in *Proceedings of IEEE International Conference on Image Processing (ICIP'94)*, (Austin, USA), pp. 122–126, 1994.

[4] M. G. Alkhansari and T. S. Huang, "A system/graph theoretical analysis of attractor coders," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, (Munich, Germany), pp. 2705–2708, 1997.

[5] M. Antonini, M. Barlaud, and P. Matjieu, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 205–220, 1992.

[6] A. A. Awainisyan, "Singularwise fractalization of gray-scaled images," in *Proceedings of the IASTED International Conference on Signal and Image Processing (SIP'97)*, (New Orleans, USA), pp. 209–214, 1997.

[7] Z. Baharav, D. Malah, and E. Karnin, "Hierarchical interpretation of fractal image coding and its application to fast decoding," in *Proceedings of the IEEE Inter-*

national Conference on Digital Signal Processing, (Nicosia, Cyprus), pp. 190–195, 1993.

[8] S. Bandyopadhyay, *Pattern Classification Using Genetic Algorithms*. PhD thesis, Indian Statistical Institute, Calcutta, India, 1998.

[9] S. Bandyopadhyay, C. A. Murthy, and S. K. Pal, "Pattern classification with genetic algorithms," *Pattern Recognition Letters*, vol. 16, pp. 801–808, 1995.

[10] B. Bani-Eqbal, "Speeding up fractal image compression," in *Still Image Compression, Volume 2418, SPIE Proceedings* (M. Rabbani, E. J. Delp, and S. A. Rajala, eds.), pp. 67–74, San Jose, USA, 1995.

[11] M. F. Barnsley, "Fractal modelling of real life images," in *The Science of Fractal Images* (H. O. Peitgen and D. Saupe, eds.), pp. 219–239, Springer Verlag, 1988.

[12] M. F. Barnsley, *Fractals Everywhere*. New York: Academic Press, 1988.

[13] M. F. Barnsley and S. G. Demko, "Iterated function systems and the global construction fractals," in *Proceedings of Royal Society, A399*, (London, UK), pp. 243–275, 1985.

[14] M. F. Barnsley, V. Ervin, D. Hardin, and J. Lancaster, "Solution of an inverse problem for fractals and other sets," in *Proceedings of the National Academy of Sciences*, (USA), 1983 (1986).

[15] M. F. Barnsley and L. P. Hurd, *Fractal Image Compression*. Wellesley: AK Peters Ltd., 1993.

[16] M. F. Barnsley and A. D. Sloan, "A better way to compress images," *Byte*, vol. 1000, pp. 215–223, 1988.

[17] K. U. Barthel, "Entropy constrained fractal image coding," *Fractals*, vol. 5, pp. 17–26, 1997.

[18] K. U. Barthel, J. Schuttemeyer, T. Voyi, and P. Noll, "A new image coding technique unifying fractal and transform coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP'94)*, (Austin, USA), pp. 112–116, 1994.

[19] F. Bartolini, V. Cappellini, S. Nerozzi, and A. Mecocci, "Recurrent neural network predictors for EEG signal compression," in *Proceedings of 1995, 20th International Conference on Acoustics,, Speech and Signal Processing, Part-5*, (Piscataway, USA), pp. 3395–3398, 1995.

[20] J. M. Beaumont, "Advances in block based fractal coding of still pictures," in *Proceedings of IEE colloquium: The Application of Fractal Techniques in image processing*, pp. 3.1–3.6, 1990.

[21] T. Bedford, F. M. Dekking, M. Breeuwer, M. S. Keane, and D. V. Schooneveld, "Fractal coding of monochrome images," *Signal Processing*, vol. 6, pp. 405–419, 1994.

[22] K. Belloulata, A. Baskurt, and R. Prost, "Fast directional fractal coding of subbands using decision directed clustering for block classification," in *Proceedings of IEEE International Conference on Acoustics, Speach and Signal Processing (ICASSP'97)*, (Munich, Germany), pp. 3121–3124, 1997.

[23] D. Bhandari, C. A. Murthy, and S. K. Pal, "Genetic algorithm with elitist model and its convergence," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 10, pp. 731–747, 1996.

[24] K. R. Boff and J. E. Lincoln, *Engineering Data Compendium : Human Perception and Performance.* OH: AAMRL, Wright-Patterson, AFB, 1988.

[25] A. Bogdan and H. Meadows, "Kohonen neural network for image coding based on iteration transform theory," in *Neural and Stochastic Methods in Image and Signal Processing* (S. S. Chen, ed.), pp. 425–436, Volume 1766 of SPIE Proceedings, 1992.

[26] R. D. Boss and E. W. Jacobs, "Archetype classification in an iterated transformation image compression algorithm," in *Fractal Image Compression: Theory and Applications* (Y. Fisher, ed.), pp. 79–90, New York: Springer Verlag, 1995.

[27] B. P. Buckles and F. E. Petry, eds., *Genetic Algorithms.* Los Alamitos: IEEE Computer Society Press, 1992.

[28] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.

[29] G. Caso, P. Obrador, and C. C. J. Kuo, "Fast method for fractal image encoding," in *Visual Communication and Image Processing, Volume 2501 of SPIE Proceedings* (L. T. Wu, ed.), pp. 583–594, Taipei, Taiwan, 1996.

[30] B. Chanda, M. K. Kundu, and V. Padmaja, "A multi-scale morphologic edge detector," *Pattern Recognition*, vol. 31, pp. 1469–1478, 1998.

[31] H. T. Chang and C. J. Kuo, "Finite state fractal block coding of images," in *Proceedings of IEEE International Conference on Image Processing (ICIP'96)*, (Lausanne, Switzerland), pp. 133–136, 1996.

[32] P. S. Chavez(Jr.), "Digital merging of landsat TM and digitised NHAP data for 1:24000 scale image mapping," *Photogrammetric Engineering and Remote Sensing*, vol. 52, pp. 1637–1646, 1986.

[33] J. S. Chen, "Fractal image compression based on visual perception," in *Human Visiom, Visual Processing and Digital Display VI, Volume 2411 of SPIE Proceedings* (B. E. Rogowitz and J. P. Allebach, eds.), pp. 92–99, 1995.

[34] C. K. Cheong, K. Aizawa, T. Saito, and M. Harori, "Structural edge detection based on fractal analysis for image compression," in *Proceedings of IEEE International Conference on Circuits and Systems*, pp. 2461–2464, 1992.

[35] C. K. Chui, *An Introduction to Wavelets.* Boston: Academic Press, 1992.

[36] L. Cieplinski, "A review on image and video coding standards," *Fundamenta Informeticae*, vol. 34, pp. 347–367, 1998.

[37] R. J. Clarke, *Transform Coding of Images.* London: Academic Press, 1990.

[38] R. J. Clarke and L. M. Linnett, "Fractals and image representation," *Electronics and Communication Engineering Journal*, vol. 4, pp. 233–239, 1993.

[39] S. Daly, "The visual difference predictor: An algorithm for the assessment of image fidelity," in *SPIE conference on Human Vision, Visual Processing and Digital Display III*, (San Jose, USA), pp. 2–15, 1992.

[40] G. Davis, "Self quantized wavelet subtree: A wavelet based theory fractal image compression," in *Wavelet Application II, Volume 2491 of SPIE Proceedings* (H. H. Szu, ed.), pp. 141–152, 1995.

[41] G. Davis, "A wavelet-based analysis of fractal image compression," *IEEE Transactions on Image Processing*, vol. 7, pp. 141–154, 1998.

[42] G. Davis, "Why fractal block coders work," in *Fractal Image Encoding and Analysis, NATO ASI Series F, vol. 159* (Y. Fisher, ed.), pp. 3–19, New York: Springer Verlag, 1998.

[43] L. Davis, *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.

[44] F. Davoine, J. Sevesson, and J. M. Chessary, "A mixed triangular and quadrilateral partition for fractal image coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP'95)*, (Washington D. C., USA), pp. 284–287, 1995.

[45] A. V. de Walle, "Merging fractal image compression and wavelet transform methods," in *NATO ASI on Fractal Image Encoding and Analysis*, (Trondheim, Norway), 1995.

[46] M. Dekking, "Fractal image coding: Some mathematical remarks on its limits and its prospects," in *Fractal Image Encoding and Analysis, NATO ASI Series F, vol. 159* (Y. Fisher, ed.), pp. 117–131, New York: Springer Verlag, 1998.

[47] W. C. Dement and N. Kleitman, "Cyclic variations in EEG during sleep and their relation to eye movements, body motility and dreaming," *Electroenceph. Clin. Neurophysiol.*, vol. 9, pp. 673–690, 1957.

[48] J. Domaszewich, S. Kullinski, and V. A. Vaishampayan, "Fractal coding versus classified transform coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP'96*, (Lausanne, Switzerland), pp. 149–152, 1996.

[49] F. Dudbridge, *Image Approximation by Self Affine Fractals*. PhD thesis, Imperial College, London, UK, 1992.

[50] F. Dudbridge, "Least-squares block coding by fractal functions," in *Fractal Image Compression: Theory and Applications* (Y. Fisher, ed.), pp. 229–241, New York: Springer Verlag, 1995.

[51] F. Dudbridge and Y. Fisher, "Attractor optimization in fractal image encoding," in *Proceedings of Fractal in Engineering*, (Arcachon, France), 1997.

[52] G. A. Edger, *Measure, Topology, and Fractal Geometry.* New York: Springer Verlag, 1990.

[53] W. H. Equitz, "A new vector quantization clustering algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, pp. 1568–1575, 1989.

[54] K. Falconer, *Fractal Geometry: Mathematical Foundations and Applications.* New York: John Wiley, 1990.

[55] J. Feder, *Fractals.* New York: Plenum Press, 1988.

[56] Y. Fisher, *Fractal Image Compression: Theory and Application.* New York: Springer Verlag, 1995.

[57] Y. Fisher, *Fractal Image Encoding and Analysis. NATO ASI Series F, vol. 159.* New York: Springer Verlag, 1998.

[58] Y. Fisher, E. W. Jacbos, and R. D. Boss, "Fractal image compression using iterated transforms," in *Image and Text Compression* (J. A. Storer, ed.), pp. 35–61, Kluwer Academic Publishers, 1992.

[59] Y. Fisher, T. P. Shen, and D. Rogovin, "Comparison of fractal methods with discrete cosine transform (DCT) and wavelets," in *Neural and Stochastic Methods in Image and Signal Processing III, Volume 2308, of SPIE Proceedings* (S. S. Chen, ed.), pp. 132–143, 1994.

[60] S. Forrest, ed., *Proceedings of 5th International Conference on Genetic Algorithms*, (San Mateo, USA), University of Illinois at Urbana-Champaign, Morgan Kaufman Publisher, 1993.

[61] B. Forte and E. R. Vrscay, "Solving the inverse problem function and image approximation using iterated function systems II: Algorithms and computations," *Fractals*, vol. 3, pp. 335–346, 1994.

[62] B. Forte and E. R. Vrscay, "Solving the inverse problem function and image approximation using iterated function systems I: Theoretical basis," *Fractals*, vol. 3, pp. 325–334, 1994.

[63] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston: Kluwer Academic Publishers, 1992.

[64] D. E. Goldberg, *Genetic Algorithms: Search, Optimization and Machine Learning*. Reading: Addison - Wesley, 1989.

[65] R. C. Gonzalez and R. R. Wood, *Digital Image Processing*. Reading: Addison-Wesley, 1993.

[66] A. Habibi, "Hybrid coding of pictorial data," *IEEE Transactions on Communications*, vol. COM-22, pp. 614–623, 1974.

[67] R. Hamzaoui, "A new decoding algorithm for fractal image compression," *Electronics Letters*, vol. 14, pp. 1273–1274, 1996.

[68] R. M. Haralick, "Digital step edge from zero crossing of second directional derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 58–86, 1984.

[69] M. E. Haziti, H. Cherifi, and D. Aboutajdine, "Complexity reduction in fractal image compression," in *Proceedings of the IASTED International Conference on Signal and Image Processing(SIP'97)*, (New Orleans, USA), pp. 245–250, 1997.

[70] H. Hinrichs, "EEG data compression with source coding techniques," *Journal of Biomedical Engineering*, vol. 13, pp. 417–423, 1991.

[71] H. L. Ho and W. K. Cham, "Attractor image coding using lapped partitioned iterated function systems," in *Proceedings of IEEE International Conference on Acoustics, Speach and Signal Processing (ICASSP'97)*, (Munich, Germany), pp. 2917–2920, 1997.

[72] B. Hurtgen and T. Hain, "On the convergence of fractal transforms," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94)*, (Adelaide, Australia), pp. 561–564, 1994.

[73] B. Hurtgen and S. F. Simon, "On the problem of convergence in fractal coding schemes," in *Proceedings of IEEE International Conference on Image Processing (ICIP'94)*, (Austin, USA), pp. 103–106, 1994.

[74] B. Hurtgen and C. Stiller, "Fast hierarchical codebook search for fractal coding of still images," in *Video Communications and PACS for Medical Applications, Volume 1977 of SPIE Proceedings* (R. A. Mattheus, A. J. Duerinckx, and P. J. V. Otterloo, eds.), pp. 397–408, 1993.

[75] J. Hutchinson, "Fractals and self similarity," *Indiana University Journal of Mathematics*, vol. 30, pp. 713–747, 1981.

[76] T. Ida and Y. Sambonsugi, "Image segmentation using fractal coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 567–570, 1995.

[77] E. W. Jacobs, Y. Fisher, and R. D. Boss, "Image compression: A study of the iterated transform method," *Signal Processing*, vol. 29, pp. 251–263, 1992.

[78] A. E. Jacquin, *Fractal Theory of Iterated Markov Operators With Applications to Digital Image Coding.* PhD thesis, Georgia Institute of Technology, Atlanta, USA, 1989.

[79] A. E. Jacquin, "A novel fractal block-coding technique for digital images," in *Proceedings of International Conference on Acoustics, Speach and Signal Processing (ICASSP'90)*, (Alberquerque, USA), pp. 2225–2228, 1990.

[80] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing*, vol. 1, pp. 18–30, 1992.

[81] A. E. Jacquin, "Fractal image coding: A review," *Proceedings of the IEEE*, vol. 81, pp. 1451–1465, 1993.

[82] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs: Prentice Hall, 1989.

[83] J. Jang and S. A. Rajala, "Segmentation based image coding using fractals and human visual system," in *Proceedings of IEEE International Conference*

on Acoustics, Speech and Signal Processing (ICASSP'90), (Alberquerque, USA), pp. 1957–1960, 1990.

[84] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, pp. 26–34, 1998.

[85] T. Kalayci, O. Ozdamar, and N. Erdol, "The use of wavelet transforms as a preprocessor for the neural network detection of EEG spikes," in *Proceedings of 1994 IEEE SOUTHEASTCON '94*, (New York), pp. 1–3, IEEE Press, 1994.

[86] S. A. Karunasekera and N. G. Kingsbury, "A distortion measure for blocking artifacts in images based on human visual sensitivity," *IEEE Transactions on Image Processing*, vol. 4, pp. 713–724, 1995.

[87] J. M. Keller, S.Chen, and R. M. Crownover, "Texture description and segmentation through fractal geometry," *Computer Vision, Graphics, and Image Processing*, vol. 45, pp. 150–166, 1989.

[88] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, pp. 1153–1160, 1981.

[89] C. S. Kim, R. C. Kim, and S. U. Lee, "Novel fractal image compression with non iterative decoder," in *Proceedings of IEEE International Conference on Image Processing (ICIP'95)*, (Washington D. C., USA), pp. 268–271, 1995.

[90] K. Kim and R. H. Park, "Image coding based on fractal approximation and vector quantization," in *Proceedings of IEEE International Conference on Image Processing (ICIP'94)*, (Austin, USA), pp. 132–136, 1994.

[91] K. Kim and R. H. Park, "Still image coding based on vector quantization and fractal approximation," *IEEE Transactions on Image Processing*, vol. 4, pp. 587–597, 1996.

[92] J. Kominek, "Codebook reduction in fractal image compression," in *Still Image Compression II, Volume 2669 of SPIE Proceedings* (R. L. stevenson, A. I. Drukarev, and T. R. Gardos, eds.), pp. 33–41, 1995.

[93] M. K. Kundu and S. K. Pal, "Edge detection based on human visual response," *International Journal of Systems and Sciences*, vol. 19, pp. 2523–2542, 1988.

[94] M. Kunt, M. Benard, and R. Leonardi, "Recent results in high-compression image coding," *IEEE Transactions on Circuits and Systems*, vol. CAS-34, pp. 1306–1336, 1987.

[95] L. Lalitha and D. D. Majumder, "Fractal based criteria to evaluate the performance of digital image magnification techniques," *Pattern Recognition Letters*, vol. 9, pp. 67–75, 1989.

[96] L. Lepsoy and G. E. Oien, "Fast attractor image encoding by adaptive codebook clustering," in *Fractal Image Compression: Theory and Applications* (Y. Fisher, ed.), pp. 177–197, New York: Springer Verlag, 1995.

[97] H. Li, M. Novak, and R. Forchheimer, "Fractal-based image sequence compression scheme," *Optical Engineering*, vol. 32, pp. 1588–1595, 1993.

[98] D. W. Lin and R. H. Park, "Fractal image coding as generalised predictive coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP'94)*, (Austin, USA), pp. 117–121, 1994.

[99] H. Lin and A. N. Venetsanopoulos, "Incorporating nonlinear contractive functions into the fractal coding," in *Proceedings of IEEE International Workshop on Intelligent Signal Processing and Communication Systems*, (Seoul, Korea), pp. 169–172, 1994.

[100] H. Lin and A. N. Venetsanopoulos, "A pyramid algorithm for fast fractal image compression," in *Proceedings of IEEE International Conference on Image Processing (ICIP'95)*, (Washington D. C., USA), pp. 596–599, 1995.

[101] H. Lin and A. N. Venetsanopoulos, "Perceptually lossless fractal image compression," in *Visual Communication and Image Processing, Volume 2727 of SPIE Proceedings* (R. Ansari and M. J. Smith, eds.), pp. 1394–1399, 1996.

[102] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, 1980.

[103] M. H. Loew, D. Li, and R. L. Pickholtz, "Adaptive PIFS model in fractal image compression," in *Medical Imaging 1996: Image Display, Volume 2707 of SPIE Proceedings* (Y. Kim, ed.), pp. 284–293, 1996.

[104] S. M. LoPresto, K. Ramchandran, and M. T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Proceedings of Data Compression Conference* (J. A. Storer and M. Cohn, eds.), IEEE Computer Society, 1997.

[105] G. Lu and T. L. Yew, "Image compression using partitioned iterated function systems," in *Image and Video Compression, Volume 2186 of SPIE Proceedings* (M. Rabbani and R. J. Safranek, eds.), pp. 122–133, 1994.

[106] N. Lu, *Fractal Imaging.* London: Academic Press, 1997.

[107] L. Lundheim, *Fractal Signal Modelling for Source Coding.* PhD thesis, The Norwegian Institute of Technology, Norway, 1992.

[108] L. Lundheim, "A discrete framework for fractal signal modeling," in *Fractal Image Compression: Theory and Applications* (Y. Fisher, ed.), pp. 79–90, New York: Springer Verlag, 1995.

[109] B. B. Mandelbort, *The Fractal Geometry of Nature.* San Francisco: Freeman, 1982.

[110] D. Marr and E. C. Hilderth, "Theory of edge detection," in *Proceedings of Royal Society, Series B*, (London, UK), pp. 187–217, 1980.

[111] G. E. McLean, "Code book edge detection," *CVGIP:Graphical Models and Image Processing*, vol. 55, pp. 48–57, 1993.

[112] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs.* Berlin: Springer Verlag, 1992.

[113] S. Mitra, S. N. Sarbadhikari, and S. K. Pal, "An MLP-based model for identifying qEEG in depression," *International Journal of Bio-Medical Engineering*, vol. 43, pp. 179–187, 1996.

[114] S. K. Mitra and C. A. Murthy, "Mathematical framework to show the existence of the attractor of the partitioned iterative function systems." *Pattern Recognition*, (Accepted).

[115] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "Edge extraction in the compressed domain using fractal reconstruction." *IEEE Transactions on Image processing*, (Communicated).

[116] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "Image edge extraction using iterative function system." *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99)*, December, 1999, Calcutta, India. (Accepted).

[117] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "A technique for image magnification using partitioned iterative function system." *Pattern Recognition*, (Accepted).

[118] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "Fractal based image coding using genetic algorithm," in *Pattern Recognition, Image Processing and Computer Vision. Recent Advances* (P. P. Das and B. N. Chatterji, eds.), pp. 86–91, New Delhi: Narosa Publishing House, 1995.

[119] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "Digital image magnification using fractal operators and genetic algorithm," in *Computational Intelligence and Applications* (P. S. Szczepaniak, ed.), pp. 250–259, Heidelberg: Physica Verlag, 1998.

[120] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "A study on partitioned iterative function systems for image compression," *Fundamenta Informaticae*, vol. 34, pp. 413–428, 1998.

[121] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "Technique for fractal image compression using genetic algorithm," *IEEE Transactions on Image Processing*, vol. 7, pp. 586–593, 1998.

[122] S. K. Mitra, C. A. Murthy, and M. K. Kundu, "Image compression and edge extraction using fractal technique and genetic algorithm," in *Soft-computing for*

Image Processing (S. K. Pal, A. Ghosh, and M. K. Kundu, eds.), Heidelberg: Physica Verlag, (Accepted).

[123] S. K. Mitra and S. N. Sarbadhikari, "EEG compression using iterated transformations and genetic algorithm," in *Proceedings of 2nd International Conference on Neural Networks and Expert Systems in Medicine and Health Care, NNESMED'96*, pp. 269–274, University of Plymouth, Plymouth, U. K., 1996.

[124] S. K. Mitra and S. N. Sarbadhikari, "Iterative function system and genetic algorithm based EEG compression," *Medical Engineering and Physics*, vol. 19, pp. 605–617, 1997.

[125] D. M. Monro, "Generalized fractal transform: Complexity issues," in *Proceedings of IEEE Data Compression Conference, (DCC'93)* (J. A. Storer and M. Cohn, eds.), pp. 254–261, 1993.

[126] D. M. Monro and F. Dudbridge, "Fractal block coding of images," *Electronics Letters*, vol. 28, pp. 1053–1055, 1992.

[127] C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithm," *Pattern Recognition Letters*, vol. 17, pp. 825–832, 1996.

[128] C. A. Murthy and S. K. Pal, "Histogram thresholding by minimising gray level fuzzyness," *Information Sciences*, vol. 60, pp. 107–135, 1992.

[129] P. Obrador, G. Caso, and C. C. J. Kuo, "A fractal based method for iterated function systems," in *Image and Video Compression, Volume 2186 of SPIE Proceedings* (M. Rabbani and R. J. Safranek, eds.), pp. 122–133, 1994.

[130] G. E. Oien, Z. Baharav, S. Lepsoy, E. Karnin, and D. Malah, "A new improved collage theorem with application to multiresolution fractal image coding," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94)*, pp. 565–568, 1994.

[131] G. E. Oien and G. Narstad, "Fractal compression of ECG signals," in *Fractal Image Encoding and Analysis* (Y. Fisher, ed.), pp. 201–226, New York: Springer Verlag, 1998.

[132] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, pp. 1277–1294, 1993.

[133] S. K. Pal and D. D. Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*. New York: John Wiley (Halsted Press), 1986.

[134] S. K. Pal, D. Bhandari, and M. K. Kundu, "Genetic algorithms for optimal image enhancement," *Pattern Recognition Letters*, vol. 15, pp. 261 – 271, 1994.

[135] S. K. Pal and P. P. Wang, eds., *Genetic Algorithms for Pattern Recognition*. Boca Raton: CRC Press, June 1996.

[136] S. K. Park and R. A. Schowengerdt, "Image reconstruction using parametric cubic convolution," *Computer Vision, Graphics, and Image Processing*, vol. 23, pp. 258–272, 1983.

[137] H. O. Peitgen and D. Saupe, *The Science of Fractal Images*. New York: Springer Verlag, 1988.

[138] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.

[139] A. P. Pentland, "Fractal based description of natural scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, pp. 661–674, 1984.

[140] A. P. Pentland and B. Horowitz, "Practical approach to fractal based image compression," in *Visual Communication and Image Processing, Volume 1605 of SPIE Proceedings* (K. H. Tzou and T. Koga, eds.), pp. 467–474, 1991.

[141] E. Polidori and J. L. Dugelay, "Zooming using iterated function systems," in *NATO ASI on Fractal Image Encoding and Analysis*, (Trondheim, Norway), 1995.

[142] D. C. Popescu and H. Yan, "MR image compression using iterated function systems," *Magnetic Resonance Imaging*, vol. 11, pp. 727–732, 1993.

[143] N. Pradhan and D. N. Dutt, "Use of running fractal dimensions for the analysis of changing patterns in electroencephalograms," *Computers in Biology and Medicine*, vol. 23, pp. 381–388, 1993.

[144] N. Pradhan and D. N. Dutt, "Data compression by linear prediction for storage and transmission of EEG signals," *International Journal of Biomedical Computing*, vol. 35, pp. 207–217, 1994.

[145] W. K. Pratt, *Digital Image Processing.* New York: John Wiley and Sons Inc., 1991.

[146] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Transactions on Communications*, vol. COM-34, pp. 1105–1115, 1986.

[147] X. Ran and N. Farvardin, "A perceptually motivated three-component image model - part I : Description of the model," *IEEE Transactions on Image Processing*, vol. 4, pp. 401–415, 1995.

[148] C. R. Rao, *Linear Statistical Inference and its Applications.* New Delhi: Wiley Eastern Limited, 1965.

[149] V. Ratnakar, E. Feig, and P. Tiwari, "Fractal based hybrid compression scheme," in *Visual Communication and Image processing, Volume 2308 of SPIE Proceedings* (A. K. Katsaggelos, ed.), pp. 448–454, 1994.

[150] E. Reusens, "Overlapped adaptive partitioning for image coding based on the theory of iterated function systems," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94)*, (Adelaide, Australia), pp. 569–572, 1994.

[151] R. Rinaldo and G. Calvango, "Image coding by block prediction of multiresolution subimages," *IEEE Transactions on Image Processing*, vol. 4, pp. 909–920, 1995.

[152] M. Ruhl, H. Hartenstein, and D. Saupe, "Adaptive partitionings for fractal image compression," in *IEEE International Conference on Image Processing (ICIP'97)*, (Santa Barbara, USA), 1997.

[153] S. N. Sarbadhikari, "A neural network confirms that physical exercise reverses EEG changes in depressed rats," *Medical Engineering and Physics*, vol. 17, pp. 579–582, 1995.

[154] D. Saupe, "Fractal image compression via nearest neighbor search," in *NATO ASI on Fractal Image Encoding and Analysis*, (Trondheim, Norway), 1995.

[155] D. Saupe and S. Jacob, "Variance based quadtree in fractal image compression," *Electronics Letters*, vol. 33, pp. 46–48, 1997.

[156] D. Saupe and M. Ruhl, "Evolutionary fractal image compression," in *Proceedings of IEEE International Conference on Image Processing (ICIP'96)*, (Lausanne, Switzerland), pp. 161–164, 1996.

[157] K. Sayood, *Introduction to Data Compression.* San Francisco: Morgan Kaufmann, 1996.

[158] J. Scharinger, F. Pichler, H. G. Feichtinger, and F. Leberl, "Comparison of lossy image compression techniques with respect to their impact on edge detection," in *Application of Digital Image Processing XIX, Vol. 2847 of SPIE Proceedings*, (Denver, USA), pp. 479–490, 1996.

[159] R. A. Schowengerdt, S. K. Park, and R. Gray, "Topics in two dimensional sampling and reconstruction of images," *International Journal of Remote Sensing*, vol. 5, pp. 333–347, 1984.

[160] J. Signes, "Reducing the codebook size in fractal image compression by geometrical analysis," in *Visual Communication and Image Processing, Volume 2727 of SPIE Proceedings* (R. Ansari and M. J. Smith, eds.), pp. 1400–1409, 1996.

[161] B. Simon, "A pyramid algorithm for fast fractal image compression," in *Proceedings of IEEE International Conference on Image Processing (ICIP'95)*, (Washington D. C., USA), pp. 278–281, 1995.

[162] W. Skarbek, "On convergence of affine fractal operators," *Image Processing and Communications*, vol. 1, pp. 33–41, 1995.

[163] W. Skarbek, "Analysis of fractal operator convergence by graph method," *Fundamenta Informaticae*, vol. 34, pp. 429–440, 1998.

[164] W. Skarbek, "Image compression using pixel neural network," in *Soft-computing for Image Processing* (S. K. Pal, A. Ghosh, and M. K. Kundu, eds.), Heidelberg: Physica Verlag, 1999.

[165] W. Skarbek and K. Ignasiak, "Asynchronous fractal operators and applications," *Image Processing and Communications*, vol. 2, pp. 3–20, 1996.

[166] W. Skarbek and K. Ignasiak, "Speedup of vector quantization by Karhunen-Loeve transform," *Image Processing and Communications*, vol. 2, pp. 35–46, 1996.

[167] A. D. Sloan, "Low-bit-rate fractal image coding," in *Visual Information Processing III, Volume 2239 of SPIE Proceedings* (F. O. Huck and R. D. Juday, eds.), pp. 210–213, 1994.

[168] M. Tanimoto, H. Ohyama, and T. Kimoto, "A new fractal image coding scheme employing blocks of variable shapes," in *Proceedings of IEEE International Conference on Image Processing (ICIP'96)*, (Lausanne, Switzerland), pp. 137–140, 1996.

[169] N. T. Thao, "A hybrid fractal-DCT coding scheme for image compression," in *Proceedings of IEEE International Conference on Image Processing (ICIP'96)*, (Lausanne, Switzerland), pp. 169–172, 1996.

[170] L. Thomas and F. Deravi, "Region-based fractal image compression using heuristic search," *IEEE Transactions on Image Processing*, vol. 4, pp. 832–838, 1995.

[171] K. Toraichi, Y. Ohtaki, Y. Ishiyama, and R. Haruki, "Compressing EEG data using adaptive function approximation," *Transactions of Institute of Electrical Engineering, Japan, Part-C*, vol. 113-C, pp. 14–20, 1993.

[172] G. Vines, "Orthogonal basis of IFS," in *Fractal Image Compression: Theory and Applications* (Y. Fisher, ed.), pp. 199–214, New York: Springer Verlag, 1995.

[173] P. D. Wakefield, D. M. Bethel, and D. M. Monro, "Hybrid image compression with implicit fractal terms," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'97)*, (Munich, Germany), pp. 2933–2936, 1997.

[174] G. K. Wallace, "The JPEG still picture standard," *Communications of the Association of Computing Machinery*, vol. 34, pp. 30–40, 1991.

[175] C. J. Wein and I. F. Blake, "On the performance of fractal compression with clustering," *IEEE Transactions on Image Processing*, vol. 5, pp. 522–526, 1996.

[176] R. Welch and M. Ehlers, "Merging multi resolutions SPOT HRV and Landsat TM data," *Photogrammetric Engineering and Remote Sensing*, vol. 53, pp. 301–303, 1987.

[177] B. E. Wohlberg and G. de Jager, "On reduction of fractal image compression encoding time," in *IEEE south African symposium on Communications and Signal processing (COMSIG'94)*, (University of Stellenbosch), pp. 158–161, 1994.

[178] B. E. Wohlberg and G. de Jager, "Fast image domain fractal compression by DCT domain block matching," *Electronics Letters*, vol. 31, pp. 869–870, 1995.

[179] C. M. Wood, "General data compression algorithm for space images using fractal techniques," in *Instrumentation in Astronomy VIII, Volume 2918 of SPIE Proceedings* (D. L. Crawford and E. R. Craine, eds.), pp. 1336–1341, 1994.

[180] N. Zhang and H. Yan, "Hybrid image compression method based on fractal geometry," *Electronics Letters*, vol. 27, pp. 406–408, 1991.

[181] Y. Zhang and L. M. Po, "Fractal color image compression using vector distortion measure," in *Proceedings of IEEE International Conference on Image Processing (ICIP'95)*, (Washington D. C., USA), pp. 284–287, 1995.

[182] X. Zhu, B. Cheng, and D. M. Titterington, "Fractal model of a one-dimensional discrete signal and its implementation," *IEE Proceedings–Vision, Image and Signal Processing*, vol. 141, pp. 318–324, 1994.

List of Publications of Author

1. S. K. Mitra, C. A. Murthy and M. K. Kundu, "Technique for Fractal Image Compression Using Genetic Algorithm" *IEEE Trans. On Image Processing*, **7**, pp. 586–593, 1998.

2. S. K. Mitra, C. A. Murthy and M. K. Kundu, "A Study on Partitioned Iterative Function System for Image Compression," *Fundamenta Informaticae*, **34**, pp 413–428, 1998.

3. S. K. Mitra, C. A. Murthy and M. K. Kundu, "Fractal Based Image Coding Using Genetic Algorithm," in *Pattern Recognition, Image Processing and Computer Vision. Recent Advances*, P. P. Das and B. N. Chatterji (Ed), Narosa Publishing House, New Delhi, pp. 86-91, 1995.

4. S. K. Mitra, C. A. Murthy and M. K. Kundu, "Image compression Using Iterative Function System and Genetic Algorithm," in *Frontiers of Radio Science Proceedings, INCURSI'96*, National Committee for International Union of Radio Science, Science City, Calcutta, p. VIII.3, January, 1996.

5. S. K. Mitra and S. N. Sarbadhikari, "EEG Compression Using Iterated Transformations and Genetic Algorithm," in *Proceedings of the 2nd International Conference on Neural Networks and Expert Systems in Medicine and Healthcare (NNESMED'96)* at School for Electronics, Communication and electrical Engineering, University of Plymouth, U. K., pp 269–274, August, 1996

6. S. K. Mitra and S. N. Sarbadhikari, "Iterative Function System and Genetic Algorithm Based EEG Compression," *Medical Engineering & Physics*, **7**, pp. 605–617, 1997.

7. S. K. Mitra and C. A. Murthy, "Mathematical Framework to show the Existence of the Attractor of the Partitioned Iterative Function Systems," *Pattern Recognition* (Accepted).

8. S. K. Mitra, C. A. Murthy and M. K. Kundu, "A technique for Image Magnification Using Partitioned Iterative Function System," *Pattern Recognition* (Accepted).

9. S. K. Mitra, C. A. Murthy and M. K. Kundu, "Image Compression and Magnification Using Fractal Technique," in *Proceedings of IEEE Region Ten Conference (TENCON 98)*, New Delhi, pp 57–60, December, 1998.

10. S. K. Mitra, C. A. Murthy and M. K. Kundu, "Fractal Image Magnification Technique," in *9th International Symposium on System- Modelling-Control (SMC 98)*, (CD Version), Zakopane, Poland, April-May, 1998.

11. S. K. Mitra, C. A. Murthy and M. K. Kundu, "Digital Image Magnification Using Fractal Operators and Genetic Algorithm," in *Computational Intelligence and Applications*, P. S. Szczepaniak (Ed), Physica Verlag, New York, pp. 250 - 259, 1999.

12. S. K. Mitra, C. A. Murthy and M. K. Kundu, "Fractal Based Image Edge Extraction," in *Proceedings of the Annual Conference of Indian Society of Information Theory and Applications (ISITA99)* (Abstract), Amritsar, India, p. 61, February, 1999.

13. S. K. Mitra, C. A. Murthy and M. K. Kundu, "Edge Extraction in the Compressed Domain Using Fractal Reconstruction," *IEEE Trans. on Image Processing* (Communicated).

14. S. K. Mitra, C. A. Murthy and M. K. Kundu, "Image Compression and Edge Extraction Using Fractal Technique and Genetic Algorithm," in *Soft- computing for Image Processing*, S. K. Pal, A.Ghosh and M. K. Kundu (Eds.), Physica Verlag (Accepted).

15. S. K. Mitra, C. A. Murthy and M. K. Kundu, "Image Edge Extraction Using Iterative Function System," in *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99)*, Calcutta, India, December, 1999.

16. S. K. Mitra, C. A. Murthy and M. K. Kundu, "A Multiscale Probabilistic Approach for Fractal Image Compression," *Signal Processing* (Communicated).